Universidade Federal do Rio de Janeiro

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Marivaldo da Silva Ferreira

CORREIO ELETRÔNICO NA MARINHA DO BRASIL: Melhores Práticas na Implementação do Lotus Notes

Rio de Janeiro

2013

Marivaldo da Silva Ferreira

CORREIO ELETRÔNICO NA MARINHA DO BRASIL:

Melhores Práticas na Implementação do Lotus Notes

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Moacyr Henrique Cruz de Azevedo, M.Sc., UFRJ, Brasil

Rio de Janeiro

2013

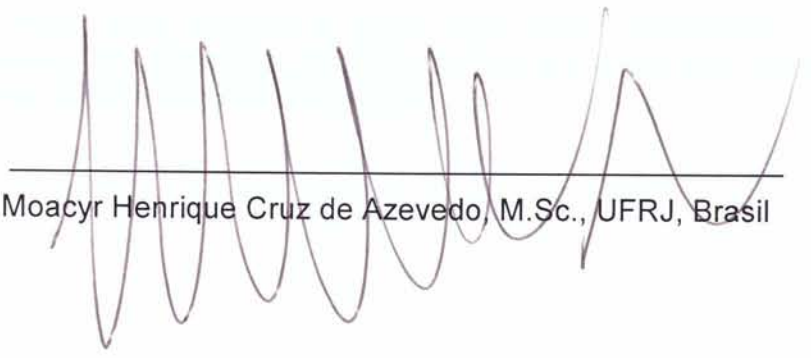
Marivaldo da Silva Ferreira

CORREIO ELETRÔNICO NA MARINHA DO BRASIL:

Melhores Práticas na Implementação do Lotus Notes

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em março de 2013.



Moacyr Henrique Cruz de Azevedo, M.Sc., UFRJ, Brasil

Dedico este trabalho a Deus, meu eterno orientador e salvador, a minha esposa e filha, minhas alegrias, e a meus pais, sem os quais eu não teria alcançado este objetivo.

AGRADECIMENTOS

Gostaria de agradecer ao meu orientador, Prof. Moacyr, pela paciência e interesse demonstrado. Agradeço também a todos os valentes que conseguiram concluir o curso MOT-2011, pela amizade e o companheirismo demonstrados nas horas difíceis.

RESUMO

FERREIRA, Marivaldo da Silva. **CORREIO ELETRÔNICO NA MARINHA DO BRASIL: Melhores Práticas na Implementação do Lotus Notes**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2013.

A infraestrutura de correio eletrônico necessária para atender a demanda de uma instituição como a Marinha do Brasil é complexa e de difícil manutenção. A Marinha possui diversos órgãos distribuídos pelo Brasil, tais como navios em constante deslocamento e edificações em terra. Possui também órgãos extraterritoriais, tais como comissões navais na América e na Europa, além de uma estação de pesquisa na Antarctica. É necessário que a informação trafegada no ambiente de correio, neste cenário tão complexo, mantenha-se sempre confiável e disponível para seus usuários. Nesse sentido, este estudo de caso procurará determinar as melhores práticas na implementação de uma infraestrutura de correio eletrônico que atenda aos requisitos de desempenho e segurança. Para tanto, serão utilizados no ambiente de testes: Sistema Operacional SUSE LINUX 11 SP2 e IBM Domino Lotus Notes versão 8.5.2.

ABSTRACT

FERREIRA, Marivaldo da Silva. **CORREIO ELETRÔNICO NA MARINHA DO BRASIL: melhores práticas na implementação do Lotus Notes**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2013.

The e-mail infrastructure necessary to meet the demands of an institution like the Navy of Brazil is complex and difficult to maintain. The Navy has several organs distributed throughout Brazil, such as ships and buildings constantly shifting ground. It also has extraterritorial bodies such as naval commissions in America and Europe, as well as a research station in Antarctica. It is necessary that the information on trafficked mail environment, on this so complex, remain reliable and always available to its users. Accordingly, this case study will determine the best practices in implementing an electronic mail infrastructure that meets the requirements of performance and security. For this purpose, will be used in the test environment: OS 11 SP2 SUSE LINUX and IBM Lotus Notes Domino version 8.5.2.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ACL	Access Control List
ARPANET	Advanced Research and Projects Agency
DMZ	Demilitarized Zone
DNS	Domain Name System
ET	Estação de Trabalho
GSI/PR	Gabinete de Segurança Institucional da Presidência da República
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
IP	Internet Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MPOG	Ministério do Planejamento Orçamento e Gestão
MTA	Mail Transfer Agent
NRPC	Notes Remote Procedure Call
OU	Organizational Unit
POP3	Post Office Protocol Version 3
RFC	Request for Comments
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

Este trabalho tem por fim verificar as melhores práticas na implementação de uma infraestrutura de correio eletrônico na Marinha do Brasil.

A motivação para elaboração deve-se ao intenso uso do correio eletrônico como instrumento de comunicação entre as diferentes organizações da Marinha, dentre as quais se destacam: navios em alto mar, comandos de distritos navais sediados em várias partes do país e comissões navais brasileiras na Europa e América, dentre outras.

Embora o correio eletrônico venha demonstrando ser uma solução adequada para permitir a comunicação entre os órgãos da Marinha, sua implementação nem sempre é executada observando-se as melhores práticas , incluindo-se aí aspectos como segurança, desempenho e confiabilidade.

Este trabalho procurará investigar quais são as melhores práticas para implementação de uma infraestrutura de correio eletrônico utilizando-se o software IBM Domino Lotus Notes.

A investigação será executada por meio da consulta aos manuais do fabricante, leitura de *Requests For Comments* (RFC) relacionada ao tema, comparação e breve descrição dos protocolos mais utilizados em correio eletrônico, e proposição da solução para o ambiente observado.

1.1 UM BREVE HISTÓRICO DO CORREIO ELETRÔNICO

A história da criação do correio eletrônico confunde-se com a da internet. Segundo o site Wikipédia [1], o correio eletrônico foi criado mesmo antes da própria internet. Na verdade, o correio eletrônico foi um dos impulsionadores da internet como a conhecemos hoje.

O criador do correio eletrônico foi um programador americano chamado Ray Tomlinson. Ele era funcionário de uma empresa contratada pelo Departamento de Defesa dos Estados Unidos da América para implementar a *ARPANET*, uma rede de comunicação de dados que ligava bases militares e centros de pesquisa americanos.

Em 1971, Ray criou um sistema para o envio e recebimento de mensagens eletrônicas entre dois *hosts* conectados nesta rede. Ele começou então a enviar mensagens para si mesmo e para seus colegas como brincadeira.

Para chegar ao seu objetivo Tomlimson somou as funcionalidades dos aplicativos SNDMSG (uma contração da expressão em inglês *send message*, ou seja, "enviar mensagem") e o *Readmail*, para leitura de correio. Mas esse sistema permitia apenas o compartilhamento de textos. O engenheiro também trabalhava em um protocolo chamado CPYNET, para transferência de arquivos entre computadores conectados em rede. Ao juntar os dois programas ele conseguiu enviar uma mensagem para seus colaboradores, anunciando sua criação.

O conceito de correio eletrônico já existia, entretanto, esse foi o primeiro sistema capaz de enviar mensagens entre diferentes nós conectados à *ARPANET*.

Em março de 1972, Ray Tomlinson escreveu o software básico de e-mail com as funções de enviar e ler, motivado pela necessidade dos desenvolvedores da *ARPANET* de ter um mecanismo de fácil coordenação. Dois anos mais tarde, um estudo indicava que setenta e cinco por cento de todo o tráfego de dados na *ARPANET* usava esse novo sistema.

1.2 NOÇÕES BÁSICAS

O correio eletrônico é uma forma de comunicação essencialmente textual, baseada no uso de redes de computadores, na internet ou intranet, que guarda semelhanças com o correio postal tradicional e com o fax.

A característica mais notável é o custo de envio de uma mensagem ser independente da distância entre o remetente e o destinatário.

Enquanto o serviço de postagem tradicional cobra pelo envio de mensagens de acordo com o tipo e volume de correspondência, distância entre remetente e destinatário, com o correio eletrônico o custo para envio de mensagens de qualquer tamanho para qualquer lugar do mundo é o mesmo.

Com o correio eletrônico também é possível enviar uma mensagem para uma pessoa ou para um grupo de pessoas, sem ter que enviar cópias para cada pessoa do grupo, bem como anexar fotos ou outros tipos de arquivos à mensagem.

O correio eletrônico também tem suas desvantagens, e a principal delas é a falta de garantia no prazo de entrega da mensagem. Para ir do computador do remetente até o do destinatário, a milhares de quilômetros de distância um do outro, uma mensagem pode levar 5 minutos ou 5 horas, dependendo das condições de tráfego na rede no momento do envio.

Ainda assim, como a mensagem resume-se, em geral, a texto simples, normalmente é possível se comunicar via *e-mail* com qualquer lugar do mundo em questão de minutos.

Atualmente, apesar do crescimento das redes sociais, as quais trouxeram recursos como *instant messenger,* o correio eletrônico ainda é um dos aplicativos de rede mais utilizado em todo o mundo, especialmente no ambiente corporativo.

2 PROTOCOLOS MAIS UTILIZADOS

Neste capítulo serão descritos os protocolos mais utilizados para o envio e recebimento de mensagens de correio eletrônico.

2.1 POST OFFICE PROTOCOL VERSÃO 3 (POP3)

O POP3 é um protocolo de acesso a correio extremamente simples e foi definido na RFC 1939 [3]. Juntamente com sua simplicidade, traz consigo várias limitações, tanto de desempenho como de segurança. É um protocolo da camada de aplicação.

A lógica de execução do protocolo é simples [2]. Ele é iniciado pelo *User Agent* (UA), ou cliente, que abre uma conexão com o servidor na porta TCP 110. Uma vez estabelecida a conexão, o protocolo passa por três fases:

1. Autorização: o UA envia usuário e senha, em texto claro, para autenticação;
2. Transação: após autenticado, o UA recupera mensagens, marca mensagens a serem apagadas e obtém estatísticas de correio; e
3. Atualização: uma vez encerrada a transação, o servidor de correio apaga as mensagens que foram marcadas para deleção e encerra a conexão.

Em uma transação POP3 o agente de usuário envia comandos e o servidor responde. Há duas respostas possíveis: +OK, para sinalizar sucesso e -ERR, para sinalizar falha na transação.

Abaixo, segue-se uma pequena ilustração destes dois comandos, utilizando-se um servidor hipotético denominado MailServer, com o protocolo POP3 habilitado:

telnet MailServer 110

+OK POP3 server ready

user João

+OK

pass 12345

+OK user successfully logged on

Esta é uma pequena descrição do funcionamento do POP3 em sua fase de autorização. Podemos perceber a simplicidade nos comandos e respostas do protocolo.

Na fase de transação o agente de usuário pode ser configurado para “ler e guardar” ou para “ler e apagar”. A sequencia de comandos emitida por este agente dependerá do modo de configuração adotado.

C: LIST

S: +OK 2 messages (320 octets)

S: 1 120

S: 2 200

S: .

...

C: LIST 2

S: +OK 2 200

...

C: LIST 3

S: -ERR no such message, only 2 messages in maildrop

O quadro 1 lista as fases e os principais comandos do POP3:

Quadro 1 – Fases e Comandos do POP3

FASE	COMANDO	DESCRIÇÃO
AUTORIZAÇÃO	USER	entra com o nome do usuário para autenticação
	PASS	entra com a senha do usuário para autenticação
	APOP	permite autenticação por meio de desafio-resposta, utilizando MD5. Permite autenticação da origem e proteção contra ataque de *replay*.
TRANSAÇÃO	STAT	lista informações da mensagem
	LIST	lista informações da mensagem de forma mais completa que o comando STAT
	RETR	retransmite a mensagem
	DELE	marca a mensagem para deleção
	NOOP	solicita uma resposta positiva do servidor. É um comando de teste.
	RSET	desmarca mensagens anteriormente marcadas para deleção pelo comando DELE
ATUALIZAÇÃO	QUIT	sai do estado de transação e entra no estado de atualização. Apaga as mensagem marcadas para deleção.

A figura 1 resume o funcionamento básico do POP3:

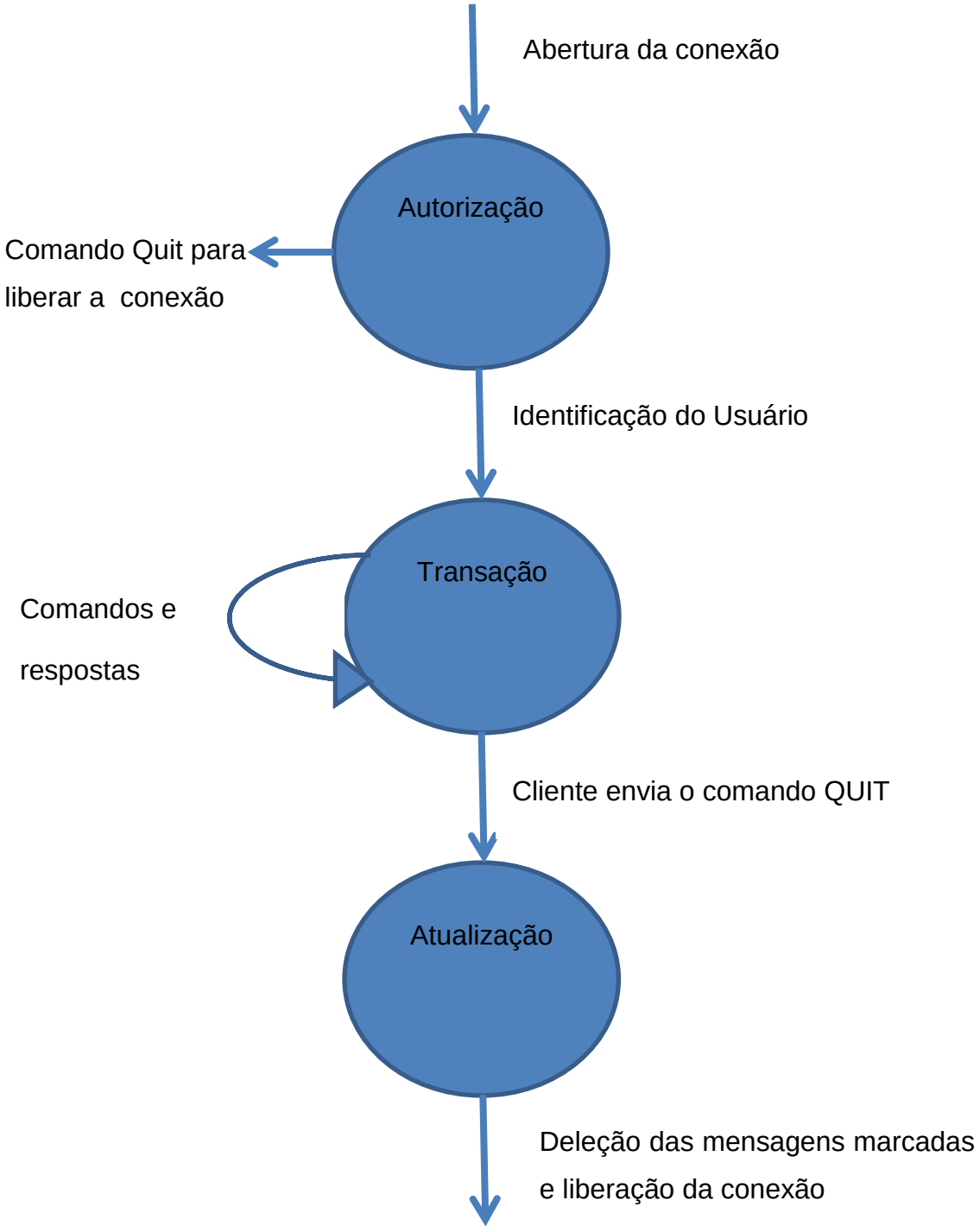


Figura 1 – Funcionamento Básico do POP3

Algumas limitações do POP3 são:

a) Impossibilidade de criar pastas no servidor de correio;
b) Impossibilidade de acesso simultâneo a uma caixa postal compartilhada por mais de um usuário;
c) Autenticação do *user agent* em texto claro, por default;

Para suplantar estas e outras deficiências foi criado o protocolo Internet Message Access Protocol (IMAP).

2.2 INTERNET MESSAGE ACCESS PROTOCOL REV1 (IMAP)

O protocolo IMAP [4] foi definido na RFC 3501. Uma de suas principais finalidades é suplantar as limitações encontradas no POP3. Quando utilizado junto com o protocolo TCP, é estabelecida uma conexão entre cliente e servidor na porta 143.

Uma vez que seja estabelecida a conexão esta pode estar em um dos seguintes estados:

1. Não autenticado: no estado não autenticado o cliente deve fornecer credenciais de autenticação, tais como usuário e senha;
2. Autenticado: neste estado o cliente é autenticado e deve selecionar uma caixa de correio para acessar, antes de poder entrar com comandos para gerenciar mensagens.
3. Selecionado: neste estado uma caixa postal foi selecionada com sucesso para manipulação de seu conteúdo.
4. Logout: neste estado a conexão está sendo encerrada pelo cliente ou pelo servidor.

Para cada estado o protocolo define comandos para interação entre o UA e o servidor de correio.

A RFC 3501 sugere diversos refinamentos e outros comandos para o protocolo IMAP, porém não faz parte do escopo deste trabalho detalhar cada um deles. O Quadro 2 lista as fases e principais comandos do IMAP.

Quadro 2 – Fases e Comandos do IMAP

ESTADO	**COMANDO**	**DESCRIÇÃO**
NÃO AUTENTICADO	AUTHENTICATE	Inicia autenticação em modo seguro
	LOGIN	Inicia autenticação em modo não seguro (senha em texto claro)
AUTENTICADO	SELECT	Seleciona uma *mail box* para acesso
	CREATE	Cria uma *mail box*
	DELETE	Marca uma *mail box* para deleção
	RENAME	Modifica o nome da *mail box*
SELECIONADO	CLOSE	Remove mensagens marcadas para deleção e retorna ao estado autenticado
	EXPUNGE	Remove mensagens marcadas para deleção
	SEARCH	Pesquisa na *mail box* de acordo com o critério solicitado
	COPY	Copia mensagens de uma *mail box* para outra

No estado LOGOUT não há comandos ou operações, por isso não está listado no quadro. É um estado designado para sinalizar a ausência de conexão entre cliente e servidor.

Algumas das vantagens do IMAP em relação ao POP3 são:

- Possibilidade de criação de pastas no servidor de correio, permitindo melhor organização do conteúdo;
- Indexação, permitindo pesquisa de mensagens;
- Permite o compartilhamento de caixas postais por um grupo;

2.3 SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

O SMTP é um protocolo da camada de aplicação [5] e foi especificado na RFC 2821.

Enquanto os protocolos POP3 e IMAP são executados no lado cliente, o SMTP tem a finalidade de “transferir mensagens de correio de servidores remetentes para servidores destinatários” [2].

O POP3 e o IMAP são utilizados para “ler” mensagens de correio eletrônico. Já o SMTP é utilizado para fazer a comunicação entre os servidores de correio eletrônico utilizando um protocolo de comunicação, normalmente TCP na porta 25.

A figura 2 ilustra o funcionamento básico do SMTP:

Figura 2 – Funcionamento Básico do SMTP

O *user agent* (UA) A quer enviar a mensagem para o *user agent* B. Para isso, deposita sua mensagem no servidor de correio A. Este, por sua vez, envia a mensagem para o servidor de correio B utilizando SMTP.

Para recuperar a mensagem do servidor de correio B, o *user agent* B executa um dos protocolos como POP3, IMAP ou HTTP, e baixa as mensagens para sua estação de trabalho (ET).

A figura 2 é uma simplificação da estrutura real de envio e recebimento de correio eletrônico. Na prática, uma mensagem de correio pode percorrer vários saltos até chegar ao seu destinatário final.

Para que isso seja possível, os servidores de envio de correio eletrônico que utilizam SMTP fazem uso da entrada MX (*Mail Exchange*) nos servidores *DNS* para localização do servidor de correio destinatário, ou do próximo salto para alcançá-lo.

2.4 NOTES REMOTE PROCEDURE CALL (*NRPC*)

O *NRPC* é um protocolo proprietário IBM Lotus Domino© utilizado para transferência de mensagens de correio eletrônico [6].

Por ser um protocolo do nível de aplicação, precisa de um protocolo do nível de transporte - normalmente TCP, porta 1352 - embora possa utilizar qualquer outro protocolo de camada 4.

É comumente empregado como protocolo de transferência de mensagens entre o cliente notes e o servidor notes, ou entre os servidores notes. A figura 3 ilustra o funcionamento do *NRPC*.

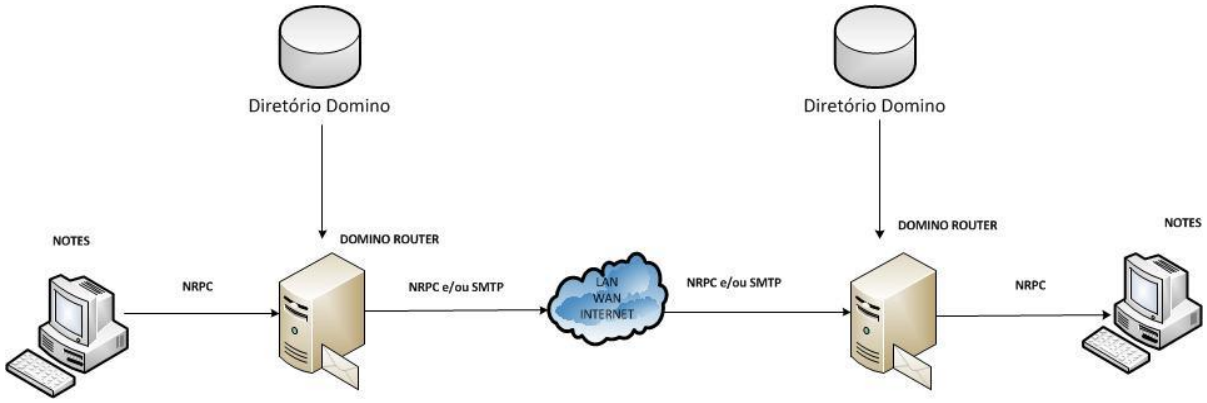


Figura 3 – Funcionamento Básico do *NRPC*

A principal diferença entre uma estrutura de envio e recebimento de mensagens que utilize *NRPC* ao invés do SMTP é que não há necessidade de utilização do serviço *DNS* para localização do próximo salto de envio.

Ao invés de consultar o *DNS*, o serviço de roteamento Domino consulta o serviço de Diretório, o qual contém a descrição e o endereço de todos os objetos existentes.

Este recurso torna a estrutura menos dependente do *DNS*, permitindo maior rapidez na entrega de mensagens e tornando o ambiente menos suscetível a erros.

3 ESTRUTURA DA REDE DOMINO LOTUS NOTES

Este trabalho está fundamentado em dois ambientes básicos: Sistema Operacional *Suse Linux Enterprise Server* (SLES) 11 SP2 e Domino Lotus Notes 8.5.2.

3.1 HISTÓRICO DO SLES 11 SP2

O nome SUSE é uma sigla alemã que significa "*Software Und System Entwicklung*" e pode ser traduzido como Desenvolvimento de Sistema e de Software [9].

O SLES foi desenvolvido baseado no *Open Suse* por uma pequena equipe liderada por Marcus Kraft e Kaindl Bernhard como principal desenvolvedor, apoiado por Joachim Schröder.

Foi lançado em 31 de outubro de 2000 como uma versão para *mainframe* IBM S/390. Em abril de 2001 o primeiro SLES para plataforma x86 foi liberado.

O SUSE Linux Enterprise Server 10 foi lançado em julho de 2006.

O SLES 11 SP1 foi lançado em maio de 2010, baseado na versão 2.6.32 do *kernel*, culminando com a versão SLES 11 SP2, cuja versão do *kernel* é a 3.0.10.

SLES 11 SP2 é nativamente um sistema operacional de 64 bits, porém aplicativos de 32 bits podem ser executados sem muitos problemas. Suporta os sistemas de arquivos: Ext3, ReiserFS, XFS, OCFS2 e Btrfs.

Como este trabalho será focado no ambiente Domino Lotus Notes, não entrará em detalhes sobre a instalação e configuração do sistema operacional SLES 11 SP2.

3.2 DEFINIÇÃO DE OBJETOS DOMINO

O Domino Lotus Notes é uma camada de software de executa sobre o sistema operacional, criando um ambiente para instalação de servidor web, servidor de correio eletrônico e servidor de aplicativos.

Para compreender o funcionamento da rede Domino Lotus Notes é necessário definir seus objetos e o funcionamento entre eles. Segue-se a definição dos principais objetos componentes da rede Domino [7].

- Diretório (*names.nsf)*: É a principal base de dados do ambiente Domino. Armazena informações dos objetos, de modo a permitir que servidores e clientes se comuniquem corretamente. É criado durante a configuração do primeiro servidor, sendo replicado em cada novo servidor instalado no domínio;
- Domínio: Não é o mesmo conceito de domínio do *Active Directory* da Microsoft. É um conjunto ou coleção de objetos (normalmente servidores e usuários), que compartilham um único Diretório. Seu objetivo principal é facilitar o roteamento de mensagens de correio. Tipicamente é definido com o nome da organização;
- Bases de dados (*filename.nsf*): *Notes Storage Facility* (nsf) é o bloco básico de construção da arquitetura notes. É a estrutura onde residem todos os dados notes. Cada arquivo .nsf é único e pode armazenar qualquer tipo de dados, incluindo aplicações, correio, diretório, gráfico, ou arquivos de áudio e vídeo;
- ID: é um arquivo associado a um objeto (normalmente objetos usuário e servidor) e que o identifica univocamente dentro do ambiente Domino. Contém o nome, o certificador, chaves pública e privada, e senha do objeto;

- Certificado: é um “carimbo” eletrônico único armazenado em um arquivo de ID, associando um nome a uma chave pública;
- ID Certificador: é um arquivo que gera o certificado para ser utilizado numa relação de confiança entre objetos do domínio;
- Rede: no ambiente Domino, este termo significa ***Domino Named Network (DNN)***, e não a infraestrutura física dos equipamentos de conexão ou os protocolos de rede. Servidores que podem se comunicar continuamente na mesma LAN/WAN utilizando o mesmo protocolo podem ser definidos como estando na mesma DNN;
- Conexão: significa “Documento de Conexão”, definido no diretório Domino. Não está associado a nenhum registro encontrado no *DNS* ou conexões físicas de rede. Trata-se de uma estrutura para permitir sincronização, replicação e comunicação entre servidores.
- *MAIL.BOX*: é uma base de dados especial que reside em cada servidor Domino usado para entrega de mensagens. A mensagem é temporariamente armazenada na *MAIL.BOX* até que a tarefa de roteamento a entregue ou transfira para o servidor de destino ou para o próximo salto.

3.3 O SERVIÇO DE MENSAGEM DO LOTUS DOMINO

Para entender como funciona o serviço de mensagem Domino, é necessário entender a correlação entre os objetos Domínio, Diretório Domino e DNN.

Se um grupo de servidores e usuários está definido no mesmo Diretório, eles estão no mesmo Domínio. Como o Diretório é uma base de dados que é replicada para todos os servidores do domínio, é essa estrutura que é utilizada pelos servidores para transferência de mensagens.

Servidores na mesma DNN podem trocar mensagns de correio automaticamente sem configurações adicionais. As DNN são utilizadas internamente pelos servidores para estabelecer as tabelas de roteamento dentro do mesmo domínio.

A figura 4 ilustra o funcionamento da rotina de roteamento no ambiente Domino utilizando-se o protocolo *NRPC*:

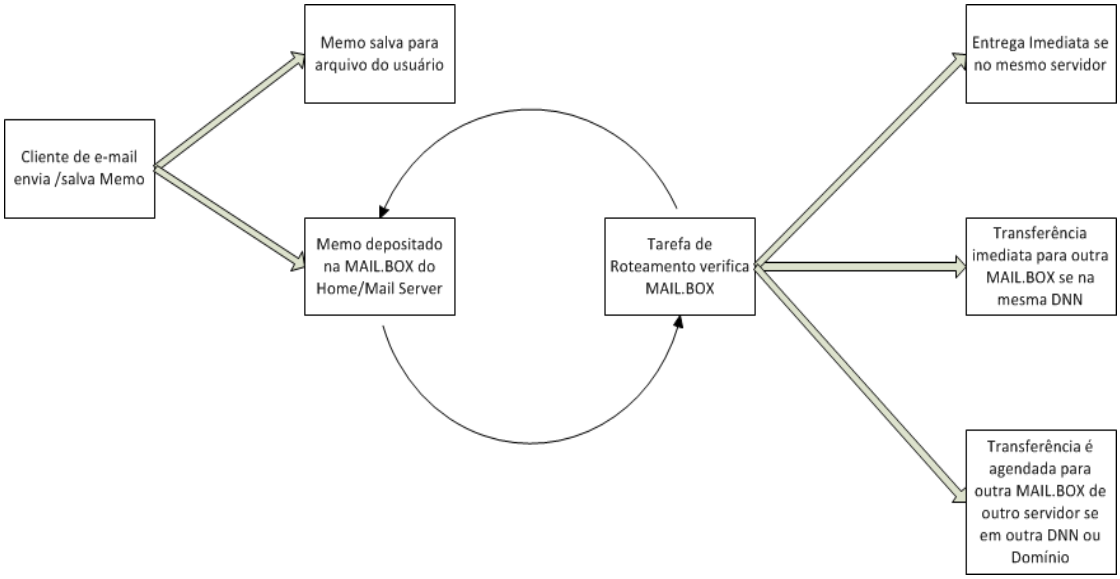


Figura 4 – Roteamento *NRPC*

Quando uma mensagem é encontrada na *MAIL.BOX*, o processo de *dispatcher* da tarefa de roteamento:

1. Entrega imediatamente a mensagem se o destino for o servidor local;
2. Transfere imediatamente a mensagem se o desttino for outro servidor na mesma DNN; ou
3. Aguarda até que o agendamento efetuado no “Documento de Conexão” atinja seu *threshold,* então transfere a mensagem para outra DNN ou Domínio. Este *threshold* é um parâmetro considerado, na maior parte das vezes, como sendo o número de mensagens armazenadas para envio.

O processo acima se repete em cada servidor até que o destino final seja alcançado, ou seja, até que a mensagem seja entregue à caixa postal do usuário final.

3.4 TOPOLOGIA DE ROTEAMENTO DE E-MAIL

A topologia escolhida para rotear mensagens de correio eletrônico estabelece como os servidores serão conectados à estrutura de rede e de que maneira estes servidores irão trocar informações.

A rede Domino utiliza a topologia para duas funções básicas:

1. Replicação: tarefa que conecta dois servidores para atualização mútua de bases de dados e sincronização do diretório; e
2. Roteamento de mensagens.

Existem duas topologias básicas para a rede Domino: *peer-to-peer* e *hub-and-spoke.* A escolha de qual delas utilizar irá depender de fatores como: número de servidores na rede, topologia da rede e velocidade dos links, dentre outros fatores.

A topologia *peer-to-peer* se adequa bem para pequeno número de servidores (até quatro) e pouco volume de tráfego.

Como ilustra a figura 5, todos os servidores se comunicam entre si, configurando uma rede *full-mesh,* com as vantagens e desvantagens inerentes a este caso.

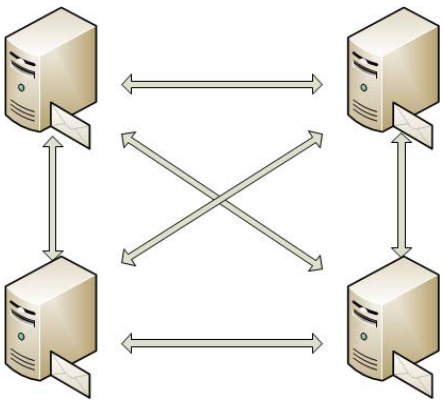


Figura 5 – Topologia *Peer-to-Peer*

Já a topologia *hub-and-spoke* é uma solução otimizada para ambientes com mais de quatro servidores [8] e grande volume de tráfego, conforme ilustra a figura 6.

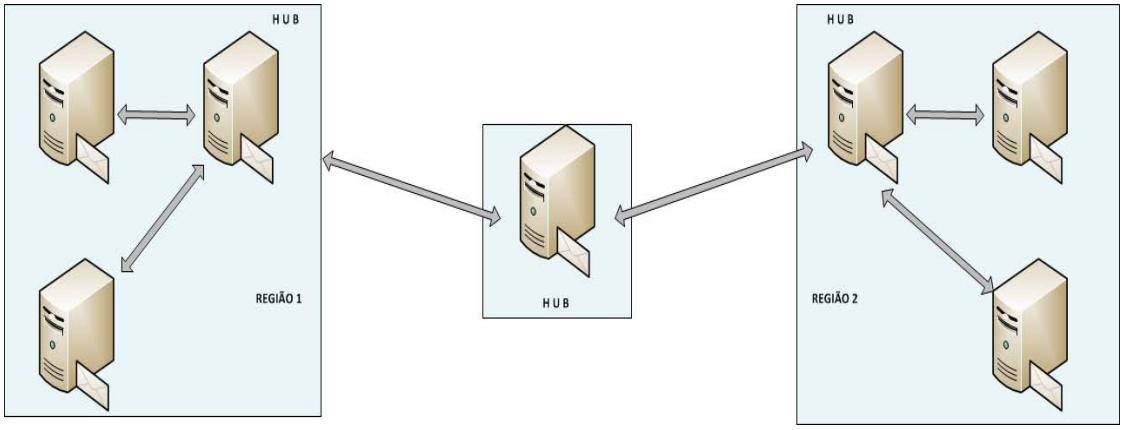


Figura 6 – Topologia *Hub-and-Spoke*

Trata-se de uma arquitetura hierárquica, onde alguns servidores concentram o tráfego, funcionando como uma espécie de *backbone*. Neste trabalho, tais servidores serão denominados “Coletores”.

4 MELHORES PRÁTICAS PARA O AMBIENTE DOMINO

Para este trabalho será considerado o ambiente de implementação das melhores práticas como sendo aquele onde se deseja obter respostas rápidas, alta eficiência e segurança na utilização de recursos.

As melhores práticas serão definidas levando-se em consideração o gerenciamento do desempenho e da segurança.

4.1 GERENCIAMENTO DE DESEMPENHO

O gerenciamento de desempenho de um ambiente como o Domino abrange não só os aspectos referentes à memória, I/O de disco ou velocidade do processador.

Embora estes aspectos sejam importantes, uma consulta rápida na documentação do software permitirá dimensionar os recursos de hardware necessários ao ambiente.

No gerenciamento de desempenho descrito neste trabalho serão pontuados os parâmetros de configuração do arquivo *notes.ini* e principais tarefas executadas no servidor .

4.1.1 Tarefas Iniciadas no *Notes.ini*

O arquivo NOTES.INI é criado no servidor Domino no momento de sua instalação. Quando da instalação do cliente notes na estação de trabalho, este arquivo também é criado localmente, afetando o ambiente de execução localmente. Este trabalho focará apenas na parte servidor.

O arquivo notes.ini contém parâmetros fundamentais para o correto funcionamento do ambiente, portanto, ao modificá-lo, é recomendável fazê-lo com muito cuidado já que uma mudança incorreta ou acidental poderá fazer com que o Domino pare de funcionar.

Existem basicamente três maneiras de editar as configurações do NOTES.INI:

1. Abrindo-o com um editor de texto comum, fazendo as alterações e salvando-as;
2. Utilizando-se o comando Set Configuration no servidor; ou
3. Criando um Documento de Definições de Configuração, alterando-o e replicando para o Diretório.

O método mais seguro é utilizar um documento de definições de configuração para modificar as configurações do servidor [10].

Abaixo está um exemplo de arquivo notes.ini para um servidor recém-instalado. Não faz parte do escopo deste trabalho pormenorizar cada um dos parâmetros listados, mas sim focar naqueles que dizem respeito ao desempenho e segurança do ambiente.

```
[Notes]
Directory=/local/notesdata
KitType=2
InstallType=EnterpriseServer
UserName=
isExpress=0
CompanyName=
NotesProgram=/opt/ibm/lotus/notes/85030/linux
ASPInstall=0
FaultRecovery_Build=Release 8.5.3
Timezone=3
DSTLAW=10,3,1,2,-1,1
SHARED_MAIL=0
DisableLDAPOnAdmin=0
Passthru_LogLevel=0
Console_LogLevel=2
DefaultMailTemplate=mail85.ntf
Preferences=32
ServerTasks=Update,Replica,Router,AMgr,AdminP,CalConn,Sched,HTTP
ServerTasksAt1=Catalog,Design
ServerTasksAt2=UpdAll
ServerTasksAt5=Statlog
TCPIP=TCP, 0, 15, 0,,32
DST=1
MailType=0
$$HasLANPort=1
Ports=TCPIP
LOG_REPLICATION=1
LOG_SESSIONS=1
```

```
KeyFileName=/local/notesdata/server.id
KeyFileName_Owner=CN=mot-1/O=linux-domino
CertifierIDFile=/local/notesdata/cert.id
MailServer=CN=mot-1/O=linux-domino
FirstServerInDomain=1
ServerKeyFileName=server.id
Domain=mot
Admin=CN=mot2011 notes/O=linux-domino
TemplateSetup=850300
Setup=850300
ServerSetup=850300
ServerKeyFileName_Owner=CN=mot-1/O=linux-domino
Log=log.nsf, 1, 0, 7, 40000
NAMELOOKUP_TRUST_DIRCAT=0
CleanSetup=1
ServerName=mot-1/linux-domino
ServerNameNative=06C006C06D6F742D312F6C696E75782D646F6D696E6F
NSF_QUOTA_METHOD=2
TRANSLOG_AutoFixup=1
TRANSLOG_UseAll=0
TRANSLOG_Style=0
TRANSLOG_Performance=2
TRANSLOG_Status=0
TTPJVMMaxHeapSize=64M
HTTPJVMMaxHeapSizeSet=1
MTEnabled=0
```

Existem inúmeros parâmetros que podem ser configurados no arquivo notes.ini a fim de aumentar o desempenho do servidor Domino.

No exemplo acima alguns parâmetros estão em negrito somente a título de destaque: a localização do arquivo executável, as tarefas que serão iniciadas automaticamente quando o servidor for ligado, o nome do servidor de correio, o domínio, o nome do administrador e o arquivo de log do ambiente.

Cada tarefa executando no ambiente Domino exige recurso computacional (memória, disco etc) do servidor. Sendo assim, o administrador precisa exercer o gerenciamento de tarefas, determinando aquelas que serão iniciadas automaticamente e aquelas que serão iniciadas em determinado horário.

O Parâmetro do *notes.ini* que permite tal gerenciamento é configurado por meio da linha *Server Tasks*.

A linha *ServerTasks* é executada durante o início do servidor, enquanto a linha *ServerTasksAt* é utilizada para iniciar tarefas em determinado horário, com o valor 0 (zero) representando meia-noite e o valor 23 (vinte e três) representando onze horas da noite.

Uma boa prática é agendar tarefas que exijam muito processamento para um momento de baixa utilização do servidor. Além disso, se um determinado recurso não vai ser utilizado é recomendável retirar o parametro referente a este recurso da linha *ServerTasks*, evitando assim que a tarefa seja iniciada e consuma recursos desnecessariamente.

No exemplo abaixo as tarefas *Router, Replica, Update, Adminp e HTTP* são iniciadas automaticamente junto com o servidor. Já as tarefas *Catalog* e *Design* são iniciadas à 01:00 hora da manhã.

ServerTasks= Router,Replica,Update,Adminp,HTTP

ServerTasksAt1=Catalog,Design

As tarefas iniciadas na linha *ServerTasks* são fundamentais para o correto funcionamento do ambiente Domino. Sua importância diz respeito não somente ao momento de inicialização, mas também à execução dos processos do ambiente.

A otimização destas tarefas faz parte do escopo deste trabalho no que diz respeito ao aspecto de desempenho. Portanto, seguem-se as recomendações para um ótimo desempenho [10]:

4.1.1.1 A Tarefa *Router*

O Domino utiliza a base de dados *MAIL.BOX* para armazenar mensagens em trânsito. A tarefa de roteamento *Router* é responsável por verificar o endereço de cada mensagem armazenada na *MAIL.BOX* e entregar a mesma para o destino

apropriado. Em um servidor de correio eletrônico esta é, sem sombra de dúvida, a tarefa mais importante.

A fim de controlar a entrega de mensagens, deve-se configurar os parâmetros abaixo para se obter um desempenho otimizado:

- *Maximum Delivery Threads*: Este parâmetro determina o número máximo de *threads* que a tarefa de roteamento pode criar para executar a entrega da mensagem no servidor local. Este é um parâmetro cujo valor ideal está entre 3 e 25.
- *Maximum Transfer Threads*: Este parâmetro determina o número máximo de *threads* quer a tarefa de roteamento pode criar para transferência de mensagens para outro servidor. O valor *default* desta variável é um thread por porta. Incrementando este valor permite à tarefa de roteamento criar mais threads para transferência de mensagens. Deve-se executar esta configuração cuidadosamente, já que aumentará a demanda por recurso de processamento.
- *Maximum Concurrent Transfer Threads:* esta configuração determina o número máximo de *threads* concorrentes por destinatário de mensagens. O valor *default* a ser configurado é o valor do parâmetro *Maximum Transfer Threads* dividido pela metade.

Para configurar os parâmetros acima, basta seguir os passos abaixo:

1. Abrir o Administrador do Domino e clicar no documento de configuração do servidor;
2. Clicar na aba *Router/SMTP*;
3. Clicar em *Restrictions and Controls*;

4. Clicar em *Delivery Controls* para configurar o parâmetro *Maximum Delivery Threads* e na aba *Transfer Control* para configurar os parâmetros *Maximum Transfer Threads* e *Maximum Concurrent Transfer Threads.*

A figura 7 ilustra o procedimento descrito acima.

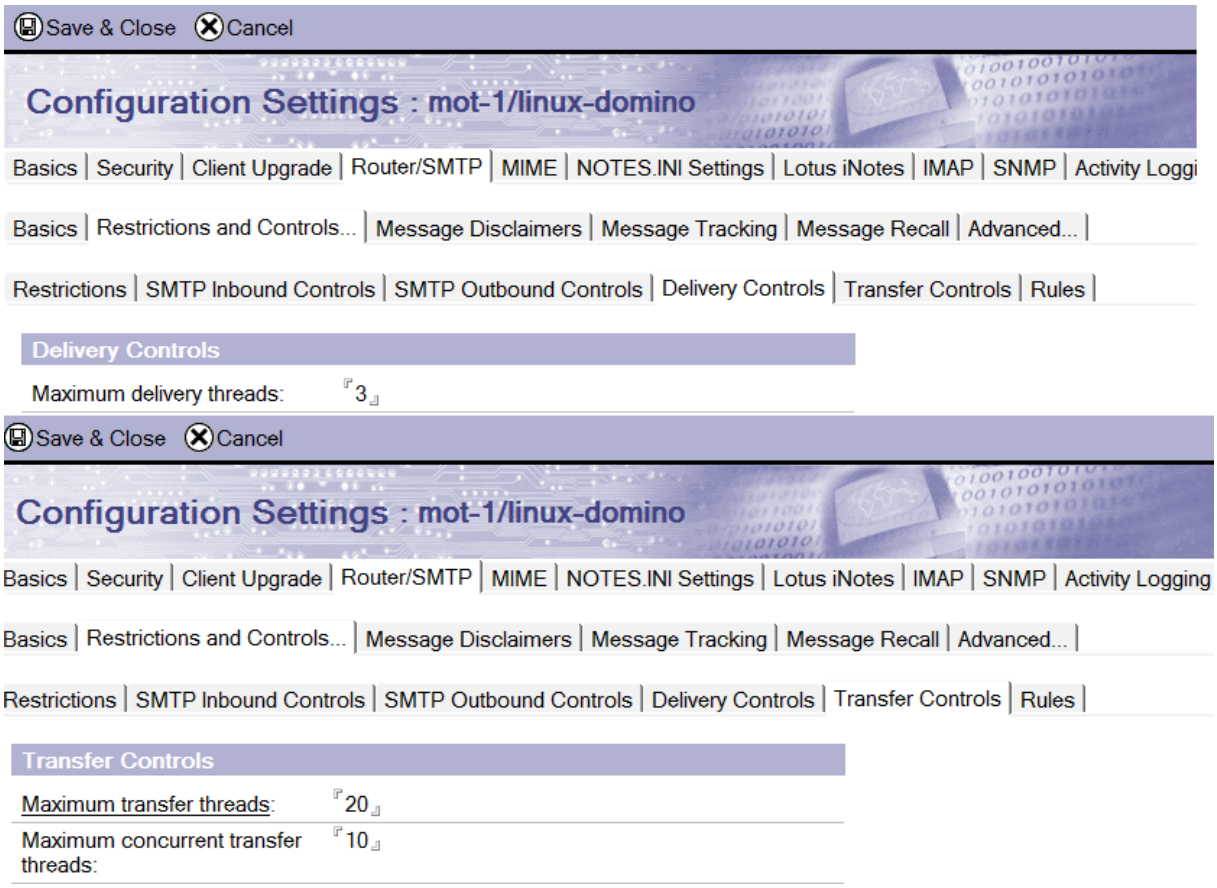


Figura 7 – Configuração de Parâmetros da Tarefa *Router*

4.1.1.2 A Tarefa *Replica*

A tarefa de replicação do Domino é responsável por gerenciar requisições de replicações agendadas. É uma tarefa de suma importância, especialmente em ambiente com múltiplos servidores, onde a consistência das informações das bases de dados deve ser mantida.

É importante ressaltar que cada replicação ocorre com um único servidor de cada vez e com uma única base de dados de cada vez. Tal entendimento é necessário para otimização do agendamento da tarefa.

Suponha que um servidor *Hub* possui um documento de conexão com um grupo de 5 servidores *Spoke* e possui um agendamento de replicação configurado com estes. Se cada um dos servidores *Spoke* possuir quarenta bases de dados em comum com o servidor *Hub*, este documento de conexão será responsável por executar duzentas replicações.

O problema descrito acima não se refere ao documento de conexão em si, mas à existência de uma única tarefa de replicação.

Uma boa prática é criar mais tarefas de replicação, levando-se em consideração o seguinte critério: uma *replicator task* para cada CPU. Para executar, basta incluir a seguinte linha no arquivo notes.ini: **replicators=x**, onde x é o número de CPUs existentes.

Como cada replicação ocorre com um único servidor de cada vez e com uma única base de dados de cada vez, adicionar tarefas de replicação permite que o Domino replique mais de uma base de dados por vez, incrementando assim o desempenho.

4.1.1.3 A Tarefa *Update*

Uma das causas mais comuns de baixo desempenho no ambiente Domino é a má configuração da tarefa *Update*, normalmente resultando em excessiva atividade.

Em uma visão simplista, pode-se descrever a estrutura do Domino como sendo composta de documentos, bases de dados (arquivos .nsf), visões , índices e tarefas que manipulam estas bases.

A tarefa *Update* é responsável pela atualização e reconstrução dos índices das bases de dados do Domino, sendo executada em *background*. Sua finalidade é otimizar as visões e bases de dados, por meio da atualização, reconstrução de

índices e visões, a fim de melhorar o desempenho e evitar que as mesmas se corrompam.

Por *default*, a tarefa *Update* (também chamada de *Indexer*) aguarda cinco segundos entre cada operação de atualização. Quando uma requisição de atualização de visão é feita, a visão somente é atualizada se ocorrer duas situações ao mesmo tempo: houver, no mínimo, vinte modificações efetuadas desde a última atualização, e se a visão foi acessada dentro dos últimos sete dias [10].

A tarefa Update mantém duas filas de trabalho: *immediate queue* (fila imediata) *e deferred queue* (fila deferida). Requisições de atualizações deferidas são mantidas suspensas por quinze minutos antes de serem processadas. Requisições da fila imediata são colocadas na situação “pendentes” até que exista um *thread* indexador disponível para processá-las.

Para verificar o desempenho da tarefa *Update*, é necessário executar o comando *show stat update* na console do servidor. O resultado do comando são estatísticas referentes à fila de atualizações, conforme ilustra a figura 8, que apresenta as seguintes estatísticas:

- *Update.DeferredList* → número de solicitações para atualização de visão ou indexação existentes na fila deferida;
- *Update.NAB.Update* → número de atualizações de visão do Diretório processadas; e
- *Update.PendingList* → número de solicitações para atualização de visão ou indexação existentes na fila imediata.

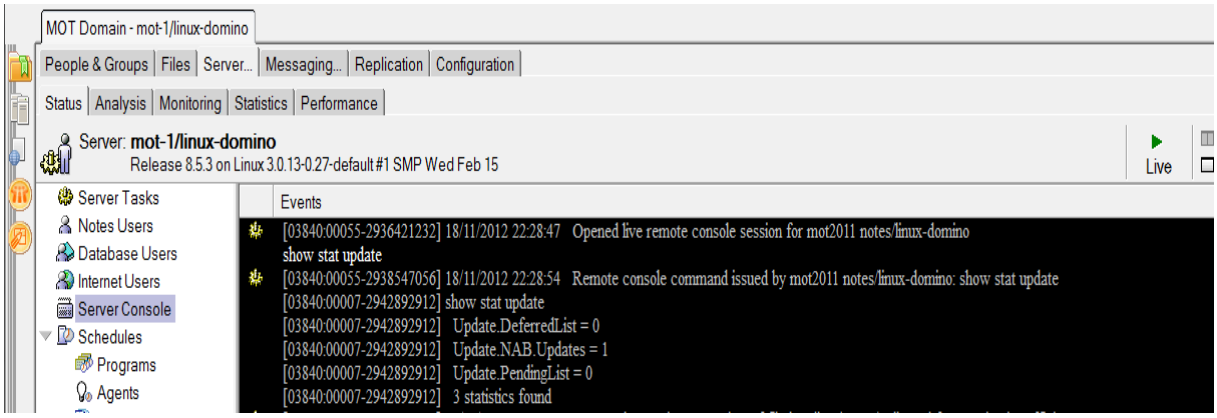


Figura 8 – Informações da Tarefa *Update*

Se o parâmetro *Update.PendingList* apresenta um número relativamente alto, isto indica necessidade de ajuste. Para definir o que seria um número alto na fila de pendências vai depender de cada caso.

É possível ajustar o desempenho da tarefa *Update* por meio dos seguintes parâmetros do *notes.ini*:

- *Update_Access_Frequency:* Este parâmetro, expresso em dias, vai permitir diminuir o número de dias necessários para que a tarefa Update seja executada sobre uma base ou visão. O *default* são sete dias. Reduzindo este número, reduzir-se-á a frequencia de atualizações em visões ou bases pouco acessadas, permitindo atualizações mais rápidas uma vez que a quantidade de informações a serem atualizadas diminuirá.
- *Update_Note_Minimum*: Este parâmetro controla quantas modificações são necessárias em uma base ou visão para que a tarefa Update seja executada sobre elas. O *default* são vinte modificações. Reduzindo este número vai forçar que a tarefa *Update* seja executada mais frequentemente. Num primeiro momento irá causar *overhead*, porém com o passar do tempo, menos *updates* serão necessários.

- *Update_Suppression_Time*: Este parâmetro controla o tempo em que as requisições de atualização são mantidas na *deferred queue* (fila deferida) antes de serem executadas. Como o valor default é de quinze minutos, diminuindo-se este valor permitirá ao Domino maior rapidez nas tarefas de *update*.

A criação de mais de uma tarefa de atualização também é uma boa prática para sistemas com mais de uma CPU, pois é recomendável a criação de tantas tarefas de *update* quanto o número de CPUs existentes no sistema.

Para executar tal alteração basta acrescentar a seguinte linha no arquivo *notes.ini:* Updaters=x, onde x é o número de CPUs do sistema [10].

4.1.1.4 A Tarefa *AdminP*

O processo denominado *AdminP* (*Administration Process*) automatiza taferas administrativas e rotineiras, tais como gerenciamento de ID, remoção de bases de dados, trocas de senhas e assim por diante.

Em ambientes com poucas tarefas administrativas não há problemas em permitir que este processo faça as atualizações no Diretório, porém, se o número de requisições ao processo *AdminP* se tornar muito alto haverá consumo excessivo de recursos e o tempo de resposta do Domino para os usuários será impactado.

Uma boa prática para gerenciamento do *AdminP* é designar um servidor centralizado dentro do domínio para ser o servidor que processará as requisições do *AdminP*.

Tarefas como trocar senha, renomear um usuário ou qualquer objeto dentro do domino (tipicamente processos administrativos), serão executadas no servidor local.

Para replicar as mudanças para todo o diretório, o processo *AdminP* executado no servidor centralizado fará isso em um horário que não seja de pico para o ambiente.

4.1.1.5 A Tarefa *HTTP*

A tarefa *HTTP* é responsável pelo consumo de uma parcela significativa dos recursos da máquina. Portanto, se o servidor Domino em questão não for utilizado como servidor *Web*, ou se os clientes acessarem o ambiente somente por meio do cliente Notes sem utilizar acesso *Web* (via *browser*) , a primeira dica é desabilitar esta tarefa.

Sendo necessário executar a tarefa, deve-se considerar a possibilidade de definir o número de *threads* utilizado pelo *web server.*

As requisições *http* são processadas por um *thread*, que por sua vez pode suportar um número limitado de conexões de rede. Uma boa prática é definir o parâmetro *Number of Active Threads* de forma a torná-lo igual ou próximo a quantidade de usuários que acessam o servidor.

A figura 9 ilustra este fato:

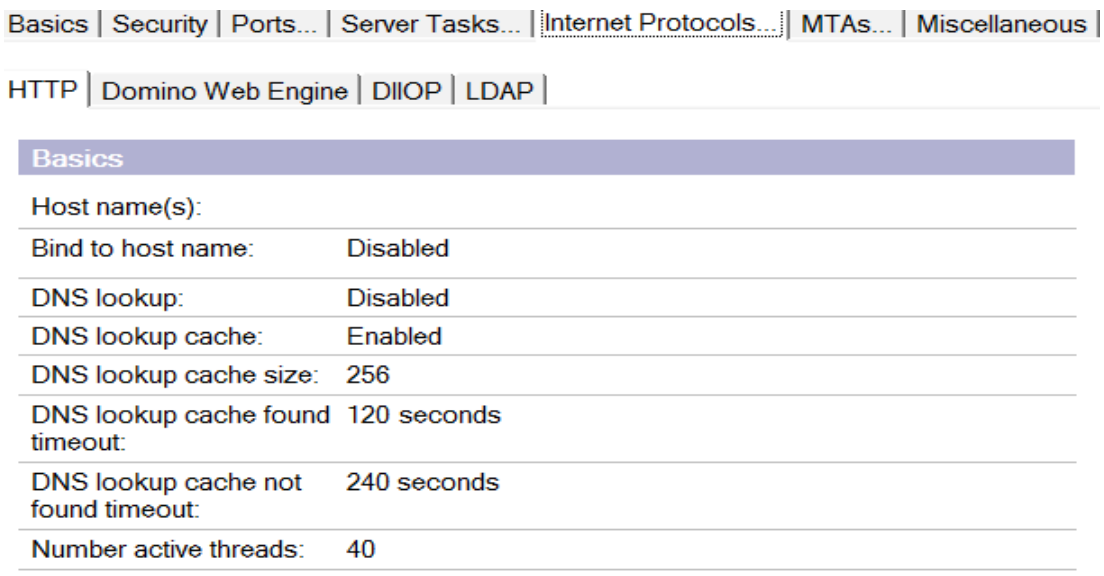


Figura 9 – Número de *Threads* da Tarefa *HTTP*

Outro aspecto importante é o volume de dados que os usuários poderão enviar para o servidor (*upload*). Para configurar este parâmetro é preciso alterar os campos *Maximum POST data* e *File compression on upload* no documento do servidor *web* conforme ilustra a figura 10:

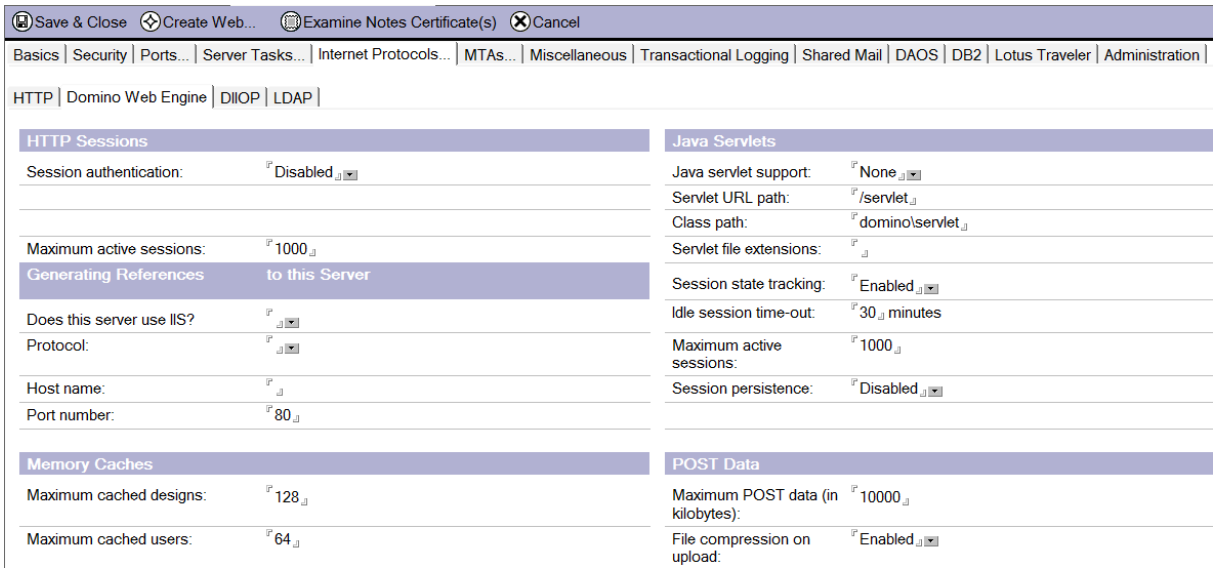


Figura 10 – Limitando o *Upload*

4.1.2. Múltiplas *MAIL.BOX*

A tarefa de roteamento (*Router)* está intimamente ligada à base *MAIL.BOX*, já que o R*outer* lê ciclicamente esta base para verificar se há mensagens a serem entregues ou transferidas.

Todos os processos do Domino que acessam a base de dados *MAIL.BOX* requerem acesso exclusivo a ela. Para isso os processos travam a base de dados a fim de evitar acesso simultâneo. Outros processos terão que aguardar até que o processo anterior termine e destrave a base de dados.

A fim de evitar esta limitação, uma boa prática é a criação de mais de uma *MAIL.BOX* no servidor. Por padrão, ao instalar um servidor Domino, é criada apenas uma *MAIL.BOX.*

Em uma estrutura do tipo *Hub-and-Spoke*, como é o caso da Marinha do Brasil, os servidores que concentram tráfego de mensagens (*hub ou* coletores) precisam ter mais de uma caixa postal criada a fim de permitir o escoamento de mensagens de maneira mais eficiente.

As vantagens da existência de mais de uma *MAIL.BOX* são:

- Permite processos concorrentes para manipulação de mensagens, aumentando o *throughput.* Enquanto um processo está lendo ou escrevendo em uma *MAIL.BOX*, outro processo pode acessar a outra *MAIL.BOX* ;
- Indiretamente, a criação de mais de uma *MAIL.BOX* disponibiliza um recurso da *failover* na eventualidade de uma delas se corromper. Se a configuração do servidor permitir, também é recomendável colocar cada *MAIL.BOX* em um disco, a fim de garantir disponibilidade em caso de falha em algum dos discos.

Para configurar mais de uma *MAIL.BOX,* basta seguir os procedimentos abaixo:

1) Abrir o Administrador do Domino, clicar na aba *Configuration*, expandir a seção *Messaging*;
2) Clicar em *Configurations;*
3) Selecionar *Configurations Settings Document* e clicar em *Edit Configuration;*
4) Clicar na aba *Router/SMTP – Basics;*
5) Entrar com o número de caixas postais desejadas, salvar e reiniciar o Domino.

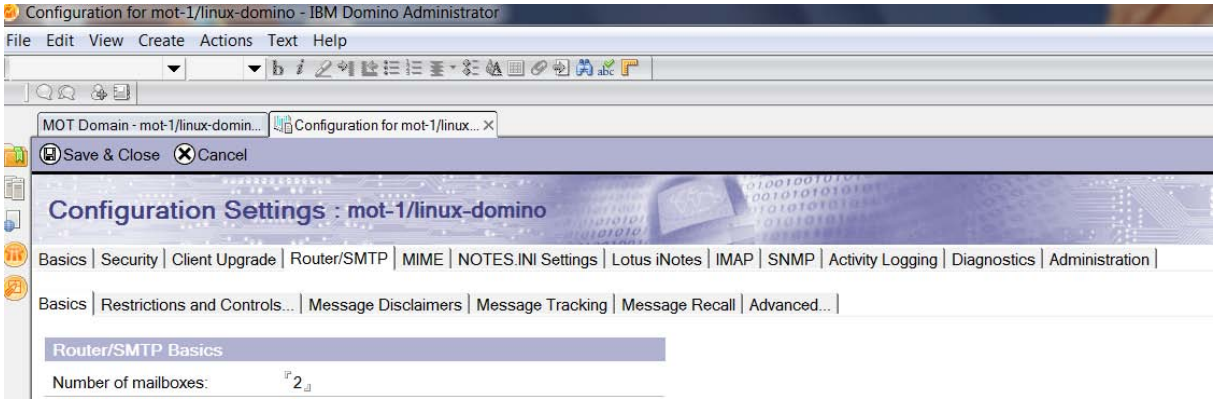


Figura 11 – Criando a Segunda *MAIL.BOX*

Após a criação da segunda *MAIL.BOX*, a estrutura do Domino ficará conforme a figura 12:

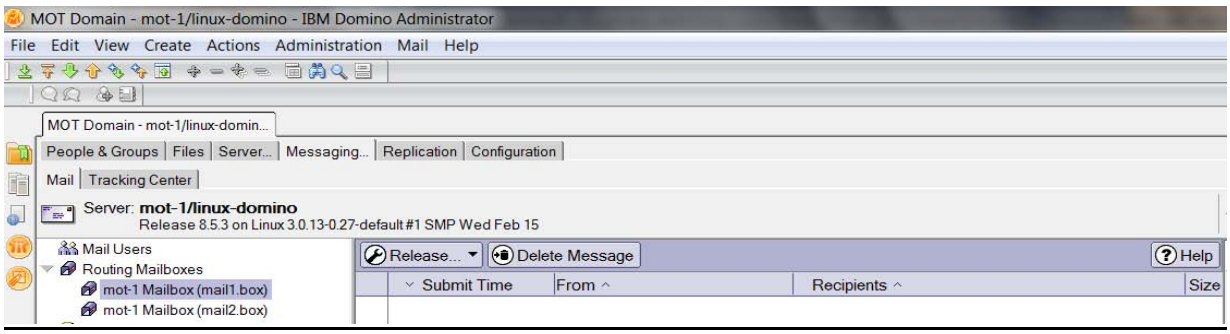


Figura 12 – Estrutura do Domino com duas *MAIL.BOX*

4.1.3 Utilitários do Domino

O servidor Domino possui alguns utilitários de linha de comando (*Updall, Compact e Fixup*) cuja execução é necessária, porém deve ser feita com muito critério.

Estes utilitários atuam executando tarefas como a atualização das visões, reconstrução dos índices e de todas as bases do Domino (*Updall*), compactação das bases, eliminando espaços vazios dentro das mesmas (*Compact*) e correção de bases e documentos corrompidos (*Fixup*).

Devem ser executados em um período fora da faixa de pico de utilização do servidor, sob o risco de tornar o ambiente extremamente lento nas respostas aos usuários.

Outro aspecto que deve ser considerado é que os utilitários *Compact* e *Updall* não removem dados durante sua execução, o que já não acontece com o *Fixup*. Este , caso encontre um elemento que não consiga corrigir, irá apagá-lo por *default*.

Os parâmetros de linha de comando de cada um dos utilitários acima podem ser encontrados na documentação do Domino e fogem ao escopo deste trabalho.

4.2 GERENCIAMENTO DA SEGURANÇA

4.2.1 Introdução

O emprego da tecnologia da informação (TI) nos processos de negócio está amplamente difundido nos dias atuais. É quase impossível conceber uma instituição governamental como a Marinha do Brasil sem que a TI esteja presente em todos os seus níveis, seja interligando suas unidades por meio de links corporativos, seja disponibilizando recursos como vídeo-conferência para comunicação operativa, ou permitindo o envio e recebimento de correio eletrônico entre seus componentes.

Juntamente com o crescimento da utilização da TI deve estar o cuidado com o correto tratamento que deve ser dado à informação que trafega ou está armazenada em meio eletrônico.

Segundo o Ministério do Planejamento, Orçamento e Gestão (MPOG) [11]: “Os dados, informações e sistemas de informação do governo devem ser protegidos contra ameaças, de forma a reduzir riscos e garantir a integridade, confidencialidade, disponibilidade e autenticidade, observando-se as normas do governo federal referentes a Política de Segurança da Informação e Comunicações, favorecendo assim, a interoperabilidade.”

A Instrução Normativa nº 1 do Gabinete de Segurança Institucional da Presidência da República (GSI/PR), de 13 de junho de 2008 [12], conceitua os requisitos de segurança como:

a) Disponibilidade: propriedade de que a informação esteja acessível e utilizável sob demanda por uma pessoa física ou determinado sistema, órgão ou entidade;
b) Integridade: propriedade de que a informação não foi modificada ou destruída de maneira não autorizada ou acidental;
c) Confidencialidade: propriedade de que a informação não esteja disponível ou revelada a pessoa física, sistema, órgão ou entidade não autorizado e credenciado;
d) Autenticidade: propriedade de que a informação foi produzida, expedida, modificada ou destruída por uma determinada pessoa física, ou por um determinado sistema, órgão ou entidade;

A fim de garantir estes requisitos, o administrador do sistema deve adaptar as configurações de fábrica do Domino às necessidades da organização, sob o risco de permitir vulnerabilidades que poderiam ser exploradas por *hackers*. É um risco instalar o servidor e pensar que configuração *default* é suficiente.

A estrutura de segurança do ambiente Domino abrange vários níveis [13], permitindo o estabelecimento de parâmetros de segurança em camadas ou em profundidade.

A figura 13 ilustra estes níveis de forma resumida:

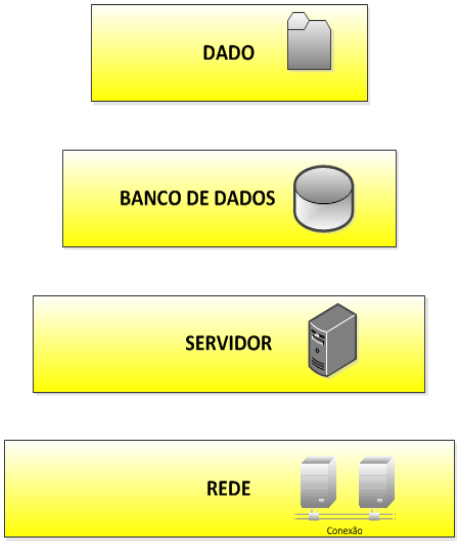


Figura 13 – Estrutura de Segurança do Ambiente Domino

A segurança começa na camada de rede. Este nível envolve a validação de quem pode acessar os recursos na rede, normalmente por meio de um usuário e senha ou outros meios de autenticação.

O Domino, nativamente, não tem envolvimento direto com este nível de segurança, porém, se a autenticação na rede e no Domino forem realizadas a partir de uma base de dados comum – utilizando-se *LDAP* por exemplo – o nível de segurança de rede passa a ser uma preocupação.

Uma vez que um usuário tenha acesso à rede, ele precisa de acesso ao Domino (nível de servidor), que é controlado por meio de autenticação e autorização.

Autenticação é a verificação da identidade do objeto que solicita acesso (pode ser um usuário, um servidor ou um processo). Existem dois tipos básicos de acesso ao ambiente Domino: por meio de um cliente instalado na estação de trabalho (ET) ou por meio do *browser*, cada um destes utilizando recursos específicos de autenticação.

Se o pedido de autenticação é realizado por meio de um cliente, é utilizado o arquivo de ID do usuário, objeto onde são armazenados o nome, a senha, e as chaves pública e privada.

Se o pedido de autenticação é feito a partir de um *browser*, ele ocorre confrontando-se um nome de usuário e senha com o campo senha de internet em um documento chamado “Pessoa” armazenado no Diretório Domino.

Autorização é a determinação daquilo que o objeto, já previamente autenticado, poderá executar no ambiente.

Após estar autenticado no servidor, o objeto necessita de direitos nas bases de dados (nível de banco de dados), os quais são dados por meio da atribuição de *Access Control List* (ACL).

Um usuário pode ter acesso total a um servidor e a todos os seus bancos de dados e, ainda assim, ser bloqueado no seu acesso a um documento ou dado em particular (nível de dado).

Portanto, observa-se que o Domino possui diversos níveis de segurança possíveis de ser implementados.

Neste trabalho serão investigados dois aspectos de segurança: utilização de política de senha e medidas anti *spam,* os quais estão relacionados ao nível de servidor.

4.2.2 Utilização de Política de Senha no Ambiente Domino

O objeto “Política” possui diversas finalidades, dentre as quais a possibilidade de criação e estabelecimento de parâmetros de senha.

Existem dois tipos básicos de política que podem ser criadas. A organizacional e a explícita. As mesmas características de segurança podem ser atribuídas tanto a uma como à outra.

A diferença básica entre elas é o escopo da aplicação: enquanto a política organizacional é aplicada a um objeto *Organizational Unit* (OU), passando a valer para todos os objetos abaixo desta OU, a política explícita pode ser aplicada a objetos individuais, tais como usuários e grupos. As duas devem ser criadas no Diretório, ou seja, é preciso direito de administrador para executar a operação.

É importante também observar a precedência entre as políticas criadas. Uma política mais específica para um determinado objeto possui precedência para execução em relação a uma política menos específica. As configurações implementadas em uma política explícita têm precedência sobre as configurações de uma política organizacional [7].

Neste trabalho será criada uma política explícita para senhas cuja complexidade é a seguinte:

a) Tamanho: mínimo de 8 e máximo de 14 caracteres;

b) Pelo menos um caracter maiúsculo;

c) Pelo menos um caracter numérico;

d) Pelo menos um caracter especial;

e) Troca obrigatória a cada semestre.

A criação desta política tem por fim forçar os usuários a utilizarem senhas que sejam difíceis de serem descobertas por ataques de força bruta e dicionário. Se a complexidade das senhas ficar sob responsabilidade dos usuários, a tendência é que eles criem senhas fáceis de serem lembradas e fáceis de serem quebradas.

Assim, a criação deste objeto permite que o administrador implemente efetivamente a política da organização quanto à utilização de senhas para autenticação no ambiente Domino.

Para criar uma política de senhas fortes para os usuários, devem-se seguir os seguintes passos:

1. Abrir o arquivo *names.nsf* com o Administrador do Domino e clicar na opção *Policies*, *Settings*, botão *Add Settings*, escolhendo a opção *Security*, conforme ilustra a figura 14:

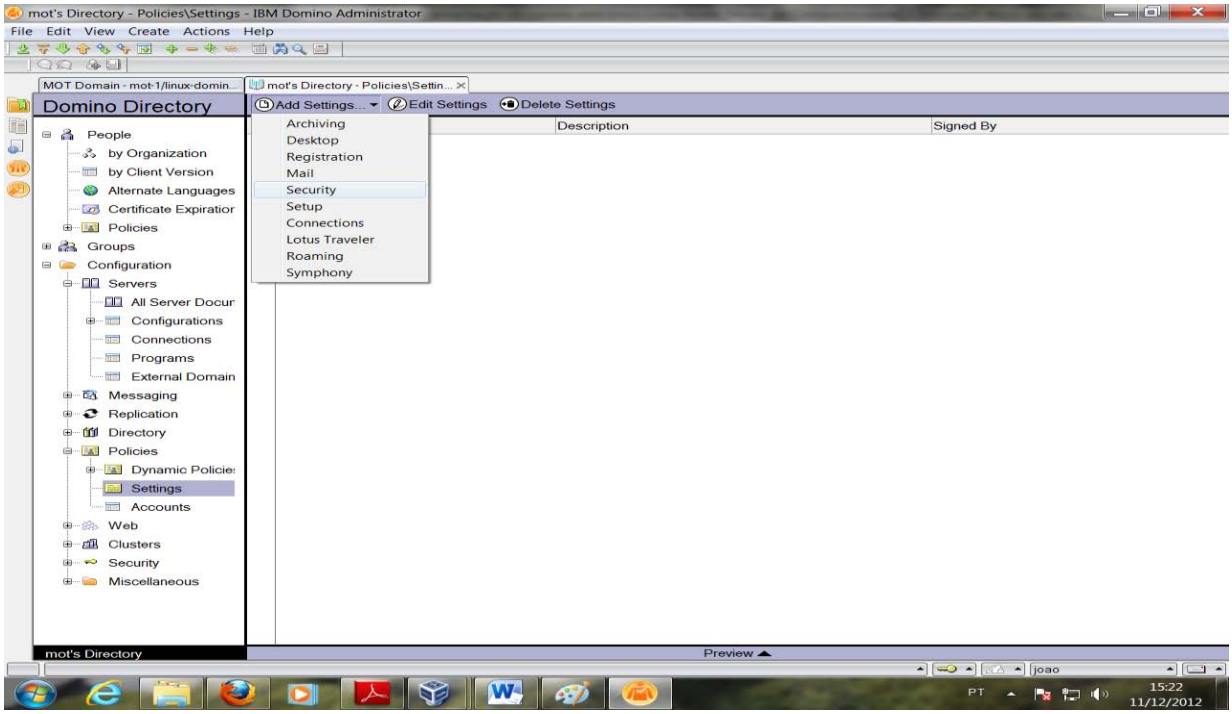


Figura 14 – Criando Documento de Configuração de Segurança

2. Preencher os campos que permitirão parametrizar a política de senha, os quais se encontram nas abas *Password Management* e *Custom Password Policy*, conforme ilustra a figura 15.

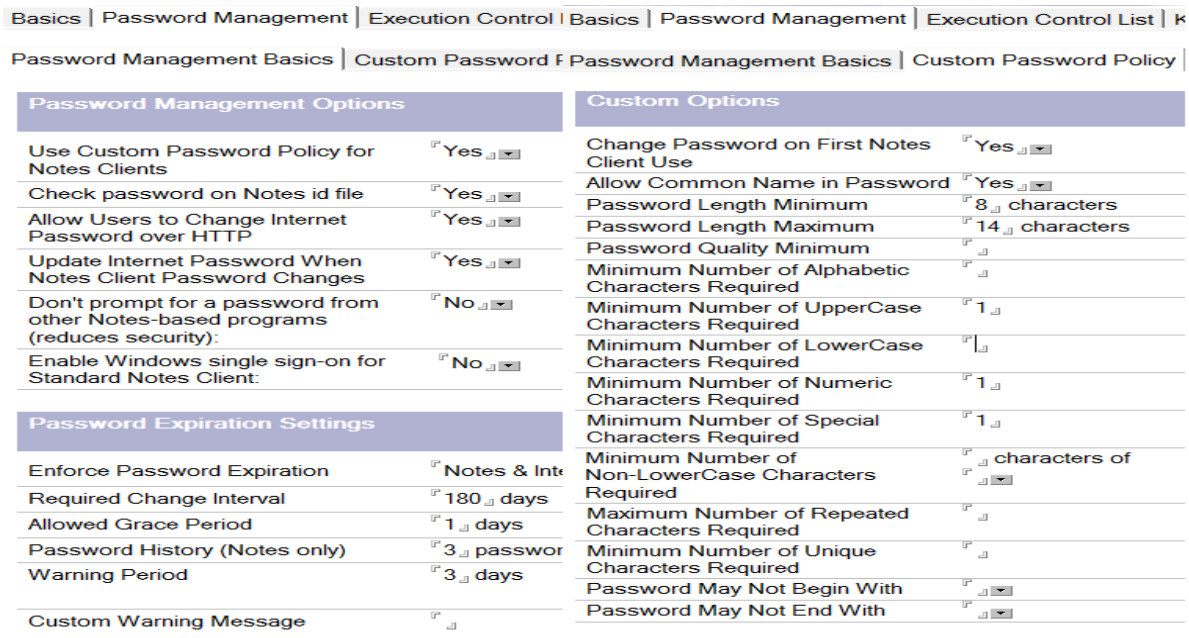


Figura 15 – Preenchimento dos Parâmetros de Senha

Os passos acima criaram o Documento de Configuração de Segurança. É preciso atribuir este documento a um objeto “Política”. Para isso, deve-se clicar na opção *Policies* no painel esquerdo, botão superior *Add Policy* e atribuir o documento no campo *Security* conforme ilustra a figura 16.

Após executar os passos anteriores, o objeto é criado no Diretório. Para atribuir a política existem dois modos: nos usuários existentes deve-se editar o objeto “Pessoa” e atribuir a política na aba *administration*; e para novos usuários deve-se atribuir a política durante a criação dos mesmos conforme ilustra a figura 16.

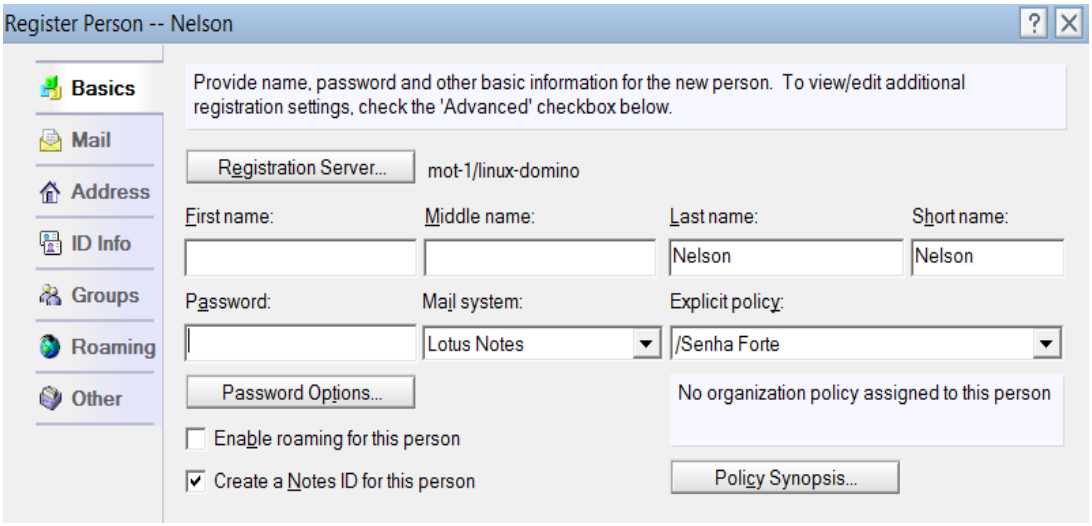


Figura 16 – Atribuindo a Política ao Usuário

A efetividade da política está ilustrada na figura 17. Quando o usuário ao qual foi atribuído a política vai se autenticar pela primeira vez no Domino, recebe um aviso que descreve quais os parâmetros de senha são necessários.

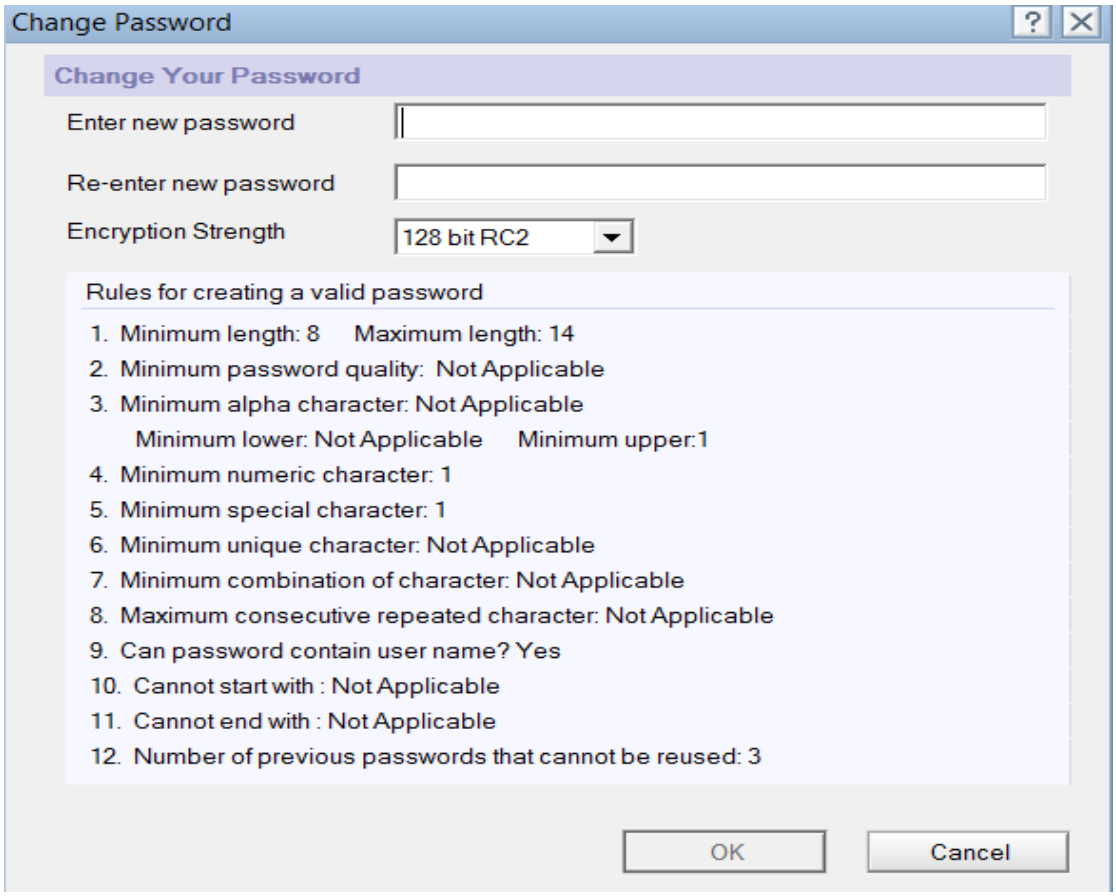


Figura 17 – Exibição da Política para o Usuário

4.2.3 Medidas anti *Spam*

Segundo o site antispam.br[14] "*Spam* é o termo usado para referir-se aos e-mails não solicitados, que geralmente são enviados para um grande número de pessoas. Quando o conteúdo é exclusivamente comercial esse tipo de mensagem é chamada de UCE (do inglês *Unsolicited Commercial E-mail*)."

A figura 18, retirada do site cert.br[16], ilustra a situação atual na internet sobre o envio e recebimento de *spam*:

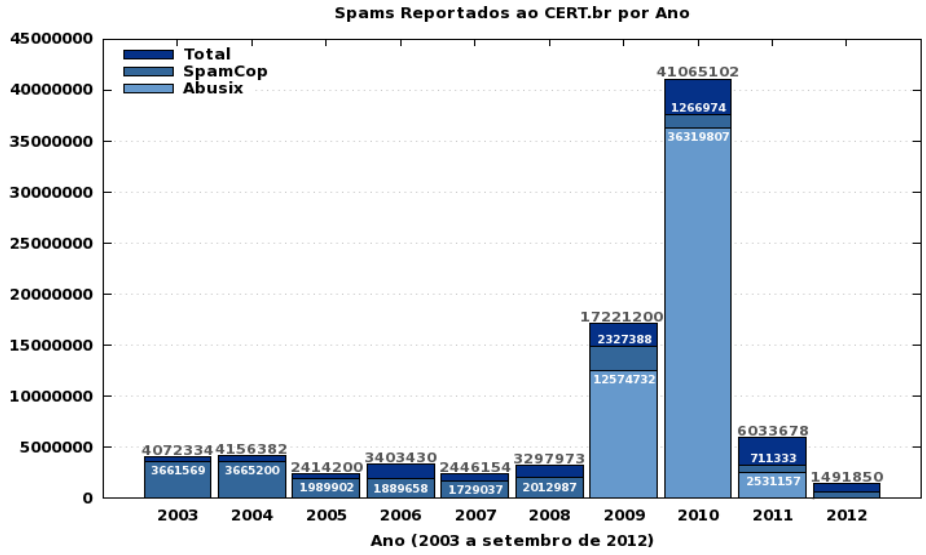


Figura 18 – *Spams* reportados ao cert.br por ano

Os prejuízos causados pelo *spam* são:

a) Risco de não recebimento de e-mails válidos: é prática comum entre administradores de correio a limitação do tamanho da caixa postal. Ao receber *spam*, uma parte do espaço que seria utilizado para recebimento de e-mails válidos é inutilizada, podendo até mesmo esgotar o espaço disponível;

b) Perda de produtividade: o tempo que o usuário gasta para abrir o e-mail, lê-lo, identificá-lo como *spam* e descartá-lo poderia ser empregado para a leitura de e-mails válidos;

c) Risco à segurança: o *spam* tem sido utilizado como veículo de propagação de código malicioso que, uma vez instalado na ET, pode abrir portas para um possível invasor, comprometendo assim a segurança das informações.

Por questões de segurança não será descrita a topologia real de correio eletrônico utilizado pela Marinha do Brasil. A estrutura deste trabalho procura buscar similaridade com o ambiente existente na Marinha do Brasil e está ilustrada na figura 19.

O servidor (1), que está numa *DMZ*, funciona como *gateway*, convertendo endereços do formato *notes NRPC* para o formato *SMTP* e vice-versa. Os servidores (2) utilizam somente *NRPC* para roteamento de mensagens internamente.

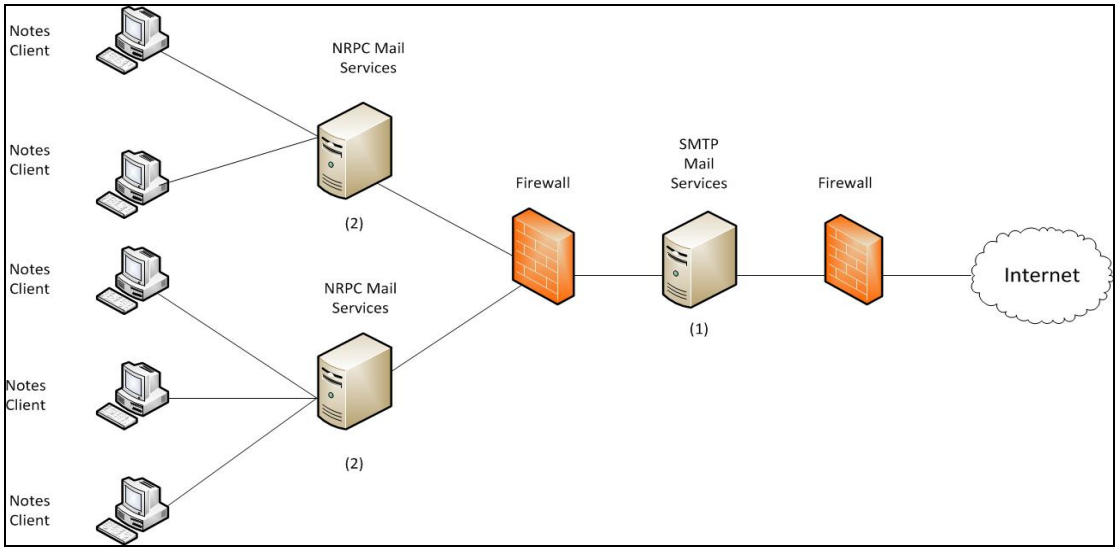


Figura 19 – Topologia Anti *Spam*

Em cada um dos dois pontos acima existem características que permitem o controle de *spam* [15]. Serão investigadas as seguintes características:

a) Filtragem de *DNS Blacklist,* configurável em (1);

b) Controle de retransmissões (*relay controls*), configurável em (1);

c) Regras de Servidor *(Server Mail Rules),* configurável em (2).

Embora estejam disponíveis, não serão investigadas as configurações referentes à caixa postal do usuário, uma vez que as características anti *spam* neste caso são implementadas pelo próprio usuário e não pelo administrador.

4.2.3.1 Filtragem de DNS *Blacklist*

Segundo o site antispam.br[17], a definição de *DNS blacklist* é a seguinte: “Trata-se de uma lista de e-mails, domínios ou endereços IP, reconhecidamente fontes de *spam*. Geralmente utiliza-se este recurso (*blacklist*) para bloquear os e-mails suspeitos de serem *Spam* no servidor de e-mails. Em alguns casos, os filtros configurados no programa leitor de e-mails também podem utilizar blacklists.”

A filtragem de *DNS blacklist* deve ser aplicada no servidor 1 da figura 19. Ela tem a vantagem de bloquear o *Spam* na *DMZ*, antes que o mesmo entre na rede interna da organização, evitando que ocupe largura de banda dos links e espaço nas caixas postais dos servidores.

Esta opção não vem habilitada por *default* no Domino. Para habilitar devem-se seguir os seguintes passos:

1. Por meio do cliente administrador do Domino, clicar na aba *Configuration*, no painel esquerdo clicar no botão *Configurations* e dar dois cliques no Documento de Configuração do Servidor para editá-lo;
2. Clicar na aba *Router/SMTP*, aba *Restrictions and Controls* e na aba *SMTP Inbound Controls*;
3. Entrar com as informações desejadas e salvar.

DNS Blacklist Filters	
DNS Blacklist filters:	Enabled
DNS Blacklist sites:	http://www.spamcop.com http://www.spamhaus.org
Desired action when a connecting host is found in a DNS Blacklist:	Log and reject message
Custom SMTP error response for rejected messages:	Conexão recusada por política de segurança

Figura 20 – Configuração de *DNS Blacklist Filters*

A figura 20 ilustra um exemplo de configuração. É importante observar os seguintes parâmetros possíveis para os campos:

- *DNS Blacklist sites*: relação de sites que o Domino irá pesquisar;
- *Desired action when a connecting host is found in a DNS Blacklist:* existem três opções para este campo:
  - ✓ *Log Only:* o Domino aceita a mensagem e grava o registro em *log*;
  - ✓ *Log and tag message:* o Domino aceita a mensagem, grava o registro em *log* e acrescenta uma *tag $DNSBLSite* a cada mensagem identificada como sendo de *blacklist.* Esta variável possibilita que o administrador possa dar tratamento personalizado às mensagens recebidas, por exemplo, por meio de um agente criado no Domino para executar tarefas específicas nas mensagens que possuem *a tag.*
  - ✓ *Log and Reject message:* o Domino rejeita a mensagem, grava o recebimento em *log* e retorna uma mensagem de erro ao remetente.
- *Custom SMTP erros response for rejected message:* permite a criação de mensagens de erro personalizadas a serem enviadas ao remetente.

4.2.3.2 Controle de Retransmissões (*Relay Controls)*

O controle de retransmissões é necessário para evitar que servidores de correio sejam identificados como *relays* abertos e sejam utilizados por *spammers* para o envio de *spam* sem serem descobertos.

Segundo o site antispam.br [18] “Relays abertos são *Mail Transfer Agents* (MTAs) que transmitem mensagens de qualquer domínio, ou mesmo só de domínios determinados, para qualquer outro, sem pedir autenticação, sem restringir (ou restringindo muito pouco) a faixa de endereços IP de origem.”

Os prejuízos imediatos para o responsável pelo *relay* aberto utilizado pelos *spammers* são: consumo de recursos do servidor e provável inclusão do domínio em *blacklists*, o que implica no não recebimento e envio de correio eletrônico.

O Domino permite restringir o *relay* por meio de configurações denominadas *Inbound Relay Controls.* Para habilitá-las devem-se seguir os seguintes passos:

1. Por meio do cliente administrador do Domino, clicar na aba *Configuration*, no painel esquerdo clicar no botão *Configurations* e dar dois cliques no Documento de Configuração do Servidor para editá-lo;
2. Clicar na aba *Router/SMTP*, aba *Restrictions and Controls* e na aba *SMTP Inbound Controls*;
3. Entrar com as informações desejadas e salvar.

A figura 21 ilustra a configuração para o bloqueio de relay pelo Domino. É importante observar que o asterisco significa “todos”.

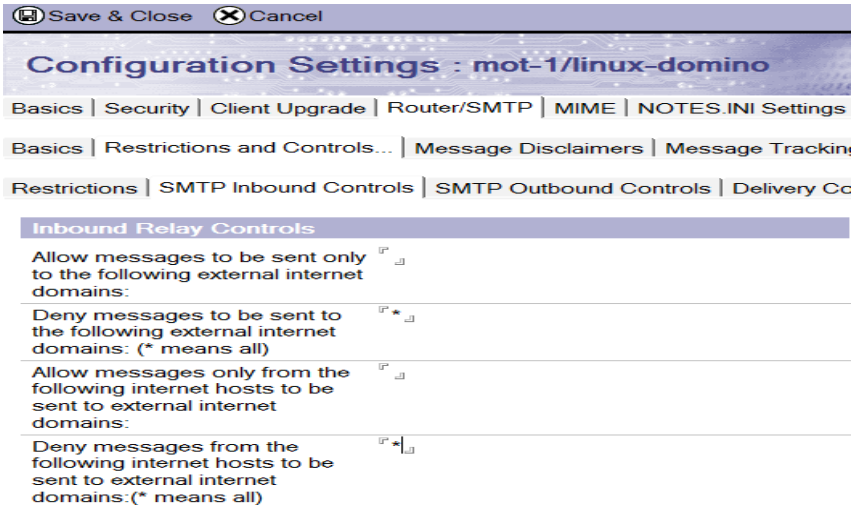


Figura 21 – Bloqueio de *Relay* no Domino

4.2.3.3 *Server Mail Rules*

A criação de regras no servidor que executa *NRPC* funciona como uma segunda linha de proteção anti *spam*. Uma vez que a mensagem de correio foi depositada na *MAIL.BOX* do servidor, as regras criadas serão executadas pela tarefa *Router* antes da entrega da mensagem na caixa postal de destino.

Para configurar regras de correio no servidor, devem-se executar os seguintes passos:

1. Por meio do cliente administrador do Domino, clicar na aba *Configuration*, no painel esquerdo clicar no botão *Configurations* e dar dois cliques no Documento de Configuração do Servidor para editá-lo;
2. Clicar na aba *Router/SMTP*, aba *Restrictions and Controls* e na aba Rules;
3. Clicar no botão *New Rule* para criar a regra desejada.

A figura 22 ilustra um exemplo de criação de regra para bloqueio de e-mails enviados com anexos cuja extensão seja de arquivo executável (exe).

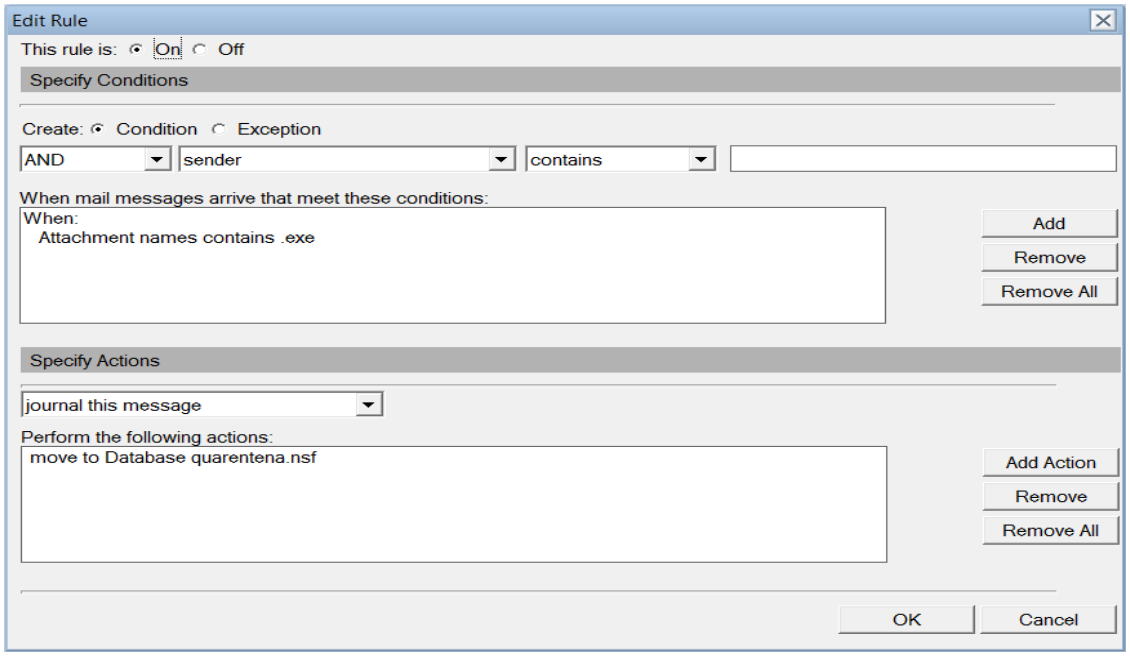


Figura 22 – Exemplo de Criação de Regra de Bloqueio

Neste caso específico, entrando uma mensagem com um anexo cuja extensão seja a de um arquivo executável, ela será movida para uma base de dados chamada quarentena, previamente criada pelo administrador.

A estratégia de criar uma base de dados para mensagens suspeitas, como no exemplo acima, possui a vantagem de permitir que o administrador possa fazer uma verificação nesta base para descobrir possíveis fontes de ataques ou até mesmo liberar uma mensagem que tenha sido identificada como falso positivo.

É possível criar inúmeras regras, porém, deve-se ter em mente que cada uma delas exige uma carga de processamento para ser executada e que a criação excessiva de regras pode impactar negativamente o desempenho do servidor.

Também é necessário lembrar que a criação de regras não é um substituto para o antivírus, cuja utilização deve ser considerada para incrementar a segurança do ambiente. O quadro 3 resume as características anti *spam* discutidas até o momento e os casos de utilização das mesmas.

Quadro 3 – Problemas Conhecidos e Soluções anti *Spam*

Problemas	*DNS Blacklist Filter*	*Inbound Relay Controls*	*Server Mail Rules*
Mensagens recebidas de um domínio conhecido como *spammer*	X		
Servidores externos utilizando seu servidor Domino como *Relay*	X	X	
Recebimento de mensagens com conteúdo suspeito			X
Servidores ou usuários internos utilizando o servidor Domino como *relay*		X	
Seu domínio foi cadastrado em blacklist		X	
Bloquear o recebimento de e-mails com *malware* anexado			X

5 CONCLUSÕES

São vários os benefícios trazidos pelo correio eletrônico: rapidez no envio e recebimento de mensagens, comodidade para o usuário e baixo custo para a Organização. Porém existem também os riscos de se utilizar um sistema complexo como este sem os devidos ajustes.

Problemas como baixo desempenho, alta latência nas respostas aos usuários, *Spam*, vírus, falta de privacidade e autenticidade, entre outros, precisam ser devidamente tratados, sob o risco de inviabilizar o uso do sistema.

O objetivo deste trabalho foi investigar as melhores práticas para implementação de uma estrutura de correio eletrônico para a Marinha do Brasil no que diz respeito aos aspectos de desempenho e segurança.

Foi observado que a infraestrutura atual de servidores de correio eletrônico não utiliza práticas recomendadas, o que gera duas consequências imediatas: subutilização do produto e vulnerabilidades exploráveis por possíveis invasores.

O trabalho foi iniciado pela revisão bibliográfica, onde foram pesquisados os protocolos POP3, IMAP, SMTP e NRPC, procurando-se verificar a evolução tecnológica e as principais características de cada um deles.

Dentre estes, o protocolo NRPC foi investigado com mais profundidade, visto ser nativo do Domino e utilizado internamente para o roteamento de mensagens entre os servidores de correio da Marinha do Brasil.

Durante tal investigação foram definidos os principais objetos do ambiente Domino Lotus Notes e suas relações no processamento de mensagens de correio eletrônico.

Na definição de objetos, percebeu-se que o Domino utiliza de forma intensa o conceito de Tarefas que, se não forem corretamente administradas, impactarão negativamente o desempenho do servidor.

Sendo assim, no que diz respeito ao desempenho, foram investigadas as características das principais tarefas que executam no ambiente a fim de identificar as configurações recomendadas por especialistas em artigos técnicos e manuais do fabricante.

No que diz respeito à segurança foram investigados os requisitos mínimos que a Marinha do Brasil, como órgão da Administração Pública Federal, deve atender para operar seu ambiente de Tecnologia da Informação e Comunicações.

Foram definidos os conceitos de confidencialidade, integridade, disponibilidade e autenticidade segundo conceitos do Governo Federal.

A fim de atender parte dos requisitos de segurança foi criada uma política de senha forte a ser utilizada para os usuários. Esta política foi implementada no ambiente de produção.

Durante a implementação foram observadas dificuldades quanto ao convencimento dos "*power users*" da organização quanto à necessidade de emprego desta política para incrementar a segurança do ambiente. O principal problema apontado foi a dificuldade na memorização da senha, agora mais complexa.

Tal problema foi minimizado por meio de palestras e disponibilização de um portal de segurança da informação para os usuários, de forma a conscientizá-los quanto à importância de se utilizar uma senha complexa. O resultado foi satisfatório, na medida em que vários órgãos da Marinha solicitaram a aplicação da política para seus servidores de correio.

Ainda no aspecto segurança, foram investigadas as características disponíveis no Domino para bloqueio de *spam*, visto ser este um meio utilizado por *hackers* para espalhar vírus e outros *malwares* que exploram vulnerabilidades.

A filtragem de domínios cadastrados em listas negras (*DNS Blacklist*) de envio de spam, o controle de retransmissões (relay controls) e a criação de regras no servidor interno foram os parâmetros configurados no ambiente de testes.

Estes parâmetros estão ligados entre si pela seguinte razão: caso o servidor de correio esteja com o serviço de retransmissão mal configurado (*relay* aberto) ele pode ser utilizado por *spammers* para envio de *spam*. Tal situação gera o risco da instituição responsável ter seu domínio inscrito automaticamente em listas negras. Estas listas serão consultadas por outros servidores de correio que bloquearão os e-mails com origem no domínio da instituição cadastrada.

Conclui-se que a adoção das boas práticas pelo administrador é um trabalho permanente de acompanhamento da evolução tecnológica do ambiente utilizado. Não é suficiente instalar o ambiente e aceitar suas configurações *default.*

Recursos cada vez mais sofisticados são disponibilizados, trazendo consigo facilidades e complexidades. Versões de software são lançadas com muita rapidez exigindo capacidade de adaptação e organização.

A implementação de boas práticas em uma estrutura já existente e que “já está funcionando” é difícil devido à resistência por parte dos que utilizam o sistema. É tarefa que exige do administrador flexibilidade para persuadi-los sobre a necessidade da mudança.

As soluções apresentadas neste trabalho não são definitivas, mas permitem ao administrador obter uma linha de ação inicial para lidar com os problemas apresentados.

Para trabalhos futuros sugere-se a investigação das características do arquivo de *log* do Domino - normalmente consultado somente quando ocorre algum problema, bem como a utilização de ferramentas automatizadas para ajuste do desempenho e segurança.

REFERÊNCIAS

[1] Disponível em http://pt.wikipedia.org/wiki/E-mail. Acessado em 26 Jul 2012.

[2] KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma Abordagem Top-Down.** São Paulo, SP. Pearson, 2010.

[3] MEYERS, J. C.M; ROSE, M. Post Office Protocol version 3. RFC 1939, disponível em http://www.ietf.org/rfc/rfc1939.txt. Acessado em 27 ago 2012.

[4] CRISPIN, M. R. Internet Access Message Protocol version 4rev1. RFC 3501, disponível em http://tools.ietf.org/html/rfc3501. Acessado em 28 ago 2012.

[5] KLENSIN, J. Simple Mail Transfer Protocol. RFC 2821. Disponível em http://www.ietf.org/rfc/rfc2821.txt. Acessado em 31 Ago 2012.

[6] Lotus Mail Administration. Disponível em http://www.waresource.com. Acessado em 10 Set 2012.

[7] IBM Lotus Domino 8 System Administration: Operating Fundamentals. Student Guide.

[8] Notes Domino Best Practices: Master Check List. Disponível em http://www-01.ibm.com/support/docview.wss?uid=swg27008523. Acessado em 17 Out 2012.

[9] Disponível em https://www.suse.com/pt-br/products/server/technical information. Acessado em 02 Nov 2012.

[10] GRAMB, T.; HARDISON, S.; JORGENSEN, C. A. B.; JAMES, L.; Domino 7 Performance Tuning - Best Practices to Get the Most Out of Your Domino Infrastructure. Disponível em http://www.redbooks.ibm.com/redpapers/pdfs/redp4182.pdf. Acessado em 03 Nov 2012.

[11] Disponível em http://eping.governoeletronico.gov.br/#s3.4. Acessado em 27 Nov 2012.

[12] Instrução Normativa nº 1 do Gabinete de Segurança Institucional da Presidência da República (GSI/PR), de 13 de junho de 2008. Disponível em http://dsic.planalto.gov.br/documentos/in_01_gsidsic.pdf, acessado em 27 Nov 2012.

[13] Overview of IBM Lotus Domino and IBM Lotus Notes Security Layers. Disponível em http://www-10.lotus.com/ldd/learnwiki.nsf/dx/Security_layers_overview.pdf/$file/Security_layers_overview.pdf. Acessado em 27 Nov 2012.

[14] TULISALO, T; CHAPPELL, T.; COLLOPY, B. A.; HANSEN, K.; KELLEHER, G.; RAMOS, M.; WALENIUS, B. Lotus Domino 6 Spam Survival Guide. Disponível em http://www.redbooks.ibm.com/redbooks/pdfs/sg246930.pdf. Acessado em 30 Nov 2012.

[15] Disponível em http://antispam.br/conceito/. Acessado em 30 Nov 2012.

[16] Disponível em http://www.cert.br/stats/spam/. Acessado em 30 Nov 2012.

[17] Disponível em http://www.antispam.br/faq/#10. Acessado em 04 Dez 2012.

[18] Disponível em http://antispam.br/admin/conf-servicos/correio/. Acessado em 04 Dez 2012.